COMMUNICATION AND COORDINATION IN WIRELESS MULTIMEDIA SENSOR AND ACTOR NETWORKS

A Thesis
Presented to
The Academic Faculty

by

Tommaso Melodia

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
August 2007

COMMUNICATION AND COORDINATION IN WIRELESS MULTIMEDIA SENSOR AND ACTOR NETWORKS

Approved by:

Professor Ian F. Akyildiz, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor John Copeland
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Faramarz Fekri
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Xiaoli Ma
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Richard Fujimoto
College of Computing
Georgia Institute of Technology

Date Approved: 06-06-2007

A Violetta

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to Prof. Ian F. Akyildiz for encouraging me to come to Georgia Tech and for his continuing guidance in the completion of this work, as well as for his valuable support as advisor during the entire Ph.D. program. I would like to sincerely thank him for seeing a potential in me and for encouraging me to become a professor, the most fulfilling and stimulating profession.

I wish to express my gratitude to all the academic members of the Electrical and Computer Engineering Department at the Georgia Institute of Technology for their excellent advice, constructive criticism, helpful and critical reviews throughout the Ph.D. program.

A special thank goes to Drs. John Copeland, Faramarz Fekri, Xiaoli Ma, and Richard Fujimoto, who kindly agreed to serve in my Ph.D. Defense Committee.

The author is grateful to his friend and colleague Dario Pompili for all the valuable work done together during the Ph.D. program. Furthermore, the author would like to thank all former and current members of the Broadband and Wireless Networking Laboratory for sharing this learning environment and experience.

Last but not least, the author is grateful to the many anonymous reviewers that with their comments greatly improved the content of the papers from which this thesis has been partly extracted.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Wireless Sensor and Actor Networks (WSANs) are distributed systems of heterogeneous devices, referred to as *sensors* and *actors*, which sense, control, and interact with the physical environment. Sensors are low-cost, low-power, multi-functional devices that communicate untethered in short distances. Actors are resource-rich devices that collect and process sensor data and consequently perform actions on the environment.

This thesis is concerned with coordination and communication problems in WSANs, in data-centric and multimedia application scenarios. First, communication and coordination problems are jointly addressed in a unifying framework for the case of static actors. A sensor-actor coordination model is proposed, based on an *event-driven partitioning* paradigm. Sensors are partitioned into different sets and each set is associated with a different actor. Data delivery trees are created to optimally react to the event and timely deliver event data with minimum energy expenditure. The optimal partitioning strategy is determined by mathematical programming, and a distributed solution is also proposed. Furthermore, the actor-actor coordination problem is formulated as an optimal task assignment problem, and a distributed solution of the problem based on an analogy with a one-shot auction is presented.

Application scenarios for WSANs with mobile actors are then studied. A location management scheme is introduced to handle the mobility of actors with minimal energy consumption for resource-constrained sensors. The proposed scheme, which is the first localization scheme specifically designed for WSANs, is shown to consistently reduce the energy consumption with respect to existing localization services for ad hoc and sensor networks. An optimal energy-aware forwarding rule is then derived for sensor-actor communication in fast varying Rayleigh channels. The proposed scheme allows controlling the delay of the data-delivery process based on power control, and reacts to network congestion by diverting traffic from congested to lightly-loaded actors. The mobility of actors is coordinated to optimally accomplish application-specific tasks, based on a nonlinear optimization model that accounts for location and capabilities of heterogeneous actors.

The research challenges for delivery of multimedia traffic in wireless sensor and actor networks are then outlined. Finally, a cross-layer communication architecture based on Ultra Wide Band communications is described, whose design objective is to reliably and flexibly *deliver QoS to multimedia applications in WSANs*, by carefully leveraging and controlling interactions among layers according to application requirements. Performance evaluation shows how the proposed solution achieves the performance objectives of wireless sensor and actor networks.

CHAPTER I

INTRODUCTION

Wireless Sensor Networks (WSN) [15] have drawn the attention of the research community in the last few years, driven by a wealth of theoretical and practical challenges. This growing interest can be largely attributed to new applications enabled by large-scale networks of small devices capable of harvesting information from the physical environment, performing simple processing on the extracted data and transmitting it to remote locations. Significant results in this area over the last few years have ushered in a surge of civil and military applications. As of today, most deployed wireless sensor networks measure scalar physical phenomena like temperature, pressure, humidity, or location of objects. In general, the applications they are designed for have small bandwidth demands, and are usually delay tolerant.

The convergence of communication and computation with signal processing and several branches of control theory such as robotics and artificial intelligence is also enabling distributed systems of embedded devices that sense, control, and interact with the physical environment. Wireless Sensor and Actor Networks (WSANs) [13] are distributed wireless systems of heterogeneous devices referred to as *sensors* and *actors*. Sensors are low-cost, low-power, multi-functional devices that communicate untethered in short distances [15]. Actors collect and process sensor data and consequently perform actions on the environment. In most applications, actors are resource rich devices equipped with high processing capabilities, high transmission power, and long battery life. The architectures, algorithms, and protocols that allow the synergic integration of sensors and actors in a wireless network are the object of this thesis.

It may be worth specifying the meaning that is attributed to the term *actor* in this thesis,

and how this differs from the more conventional notion of *actuator*. An actuator is a device to convert an electrical control signal to a physical action, and may be used for flow-control valves, pumps, positioning drives, motors, switches, relays and meters. It constitutes the mechanism by which an agent acts upon the physical environment. From the perspective considered in this thesis, however, an actor, besides being able to act on the environment by means of one or several actuators, is also a *single and separated network entity* that performs networking-related functionalities, i.e., receive, transmit, process, and relay data. For example, a robot may interact with the physical environment by means of several motors and servo-mechanisms (actuators). However, from a networking perspective, the robot constitutes a single entity, which we refer to as actor.

In WSANs, a possibly large number of sensor nodes, i.e., on the order of hundreds or thousands, are randomly deployed in a target area to perform a collaborative sensing task. Such a dense deployment may not be necessary for actors, since actors are in many practical scenarios sophisticated devices with higher capabilities that can act on large areas.

Research on WSANs presents several overlaps with other research disciplines, and aims at complementing and integrating them. For example, *Distributed Robotics* [24] has been a hot research topic since the mid 90's. In distributed robotics each task, instead of being accomplished by a single, isolated robot, is performed by a *team of collaborating robots*. Information about the surrounding environment is usually gathered by *onboard sensors*, and team members exchange sensor information to move or perform actions (e.g., collaborate to manipulate heavy objects). As opposed to a single robot, a team of robots can perceive the environment from *multiple disparate viewpoints*. In WSANs, the ability of the actors to perceive the environment can be pushed one step further: a dense spatio-temporal sampling of the environment, provided by a pre-deployed sensor network, can be exploited by the whole team of actors, thus increasing the ability of the team to accurately interact with the physical environment. This also enables a new design perspective, as it makes possible to build simpler, less expensive robots easier to maintain and debug that

can accomplish the goal reliably through cooperation with the sensor network. Moreover, multimedia content gathered by sensors can be used to provide the team of actors with accurate vision from multiple perspectives, while as of today collaborating actors mostly rely on expensive onboard cameras.

In WSANs, the collaborative operation of sensors enables the *distributed sensing* of a physical phenomenon. After sensors detect an event that is occurring in the environment, the event data is distributively processed and transmitted to the actors, which gather, process, and eventually reconstruct the characteristics of the event. The process of establishing data paths between sensors and actors is referred to as *sensor-actor coordination* [13]. Once the event has been detected, actors coordinate to reconstruct it, to estimate its characteristics, and make a collaborative decision on how to perform the action. This process is referred to as *actor-actor coordination* [13]. As a result, the operation of a WSAN can be thought of as an event-sensing, communication, decision, and acting loop. WSANs can be seen as a distributed control system designed to timely react to sensor information with an effective action. For this reason, *real-time coordination and communication* is an important concern in WSANs to guarantee timely execution of the right actions. The *energy efficiency* of network communications is also crucial, since sensors are resource-constrained nodes with limited battery lifetime [15]. Furthermore, network protocols and algorithms should be *scalable* and *localized*, as the number of nodes can be arbitrarily high.

For these reasons, in WSANs it is required to deliver event data in an energy-efficient way while abiding by real-time delay constraints, i.e., the operating point of the network should be cooperatively adjusted by sensors and actors to minimize the energy consumption, given some requirements on the acceptable delay. Another primary requirement of WSANs is *scalability*, i.e., the property of performing operations efficiently when the number of network devices increases.

Along with scalar data typically measured by sensors, multimedia content gathered by video and audio sensors pre-deployed on the environment can be used to provide the

team of actors with accurate vision from multiple perspectives, while as of today collaborative actors mostly rely on expensive onboard cameras. The recent availability of inexpensive hardware such as CMOS cameras and microphones that are able to ubiquitously capture multimedia content has fostered the development of multimedia communications over wireless sensor networks [14], thus paving the way for networks of wirelessly interconnected devices that allow retrieving video and audio streams, still images, and scalar sensor data. With rapid improvements and miniaturization in hardware, a single sensor device can be equipped with audio and visual information collection modules. In addition to the ability to retrieve multimedia data, wireless multimedia sensor and actor networks (WMSANs) will be able to store, process in real time, correlate and fuse multimedia data originated from heterogeneous sources, and perform actions on the environment based on the content gathered. Many applications require the sensor network paradigm to be rethought in view of the need for mechanisms to deliver multimedia content with a certain level of quality of service (QoS). Since the need to minimize the energy consumption has driven most of the research in sensor networks so far, mechanisms to efficiently deliver application-level QoS, and to map these requirements to network-layer metrics such as latency, have not been primary concerns in mainstream research on sensor networks.

1.1 Applications of WSANs

Several applications for WSANs are concerned with *enhancing and complementing existing sensor network applications*. In these applications, the performed actions serve the purpose of enhancing the operation of the sensor network by enabling or extending its monitoring capability. For example, mobile actors can accurately deploy sensors [108], enable adaptive sampling of the environment [82], pick up data from the sensors when in close range, buffer it, and drop off the data to wired access points [100], perform energy harvesting [90], or enhance the localization capabilities of sensors [54].

Conversely, our research is concerned with applications where actors are part of the

network itself and perform actions based on the information gathered by sensors. It is envisioned that WSANs will be an integral part of systems such as battlefield surveillance, nuclear, biological or chemical attack detection, home automation, and environmental monitoring [13]. For example, in fire detection applications, sensors can relay the exact origin and intensity of the fire to water sprinkler actors that will extinguish the fire before it spreads. Moreover, sensors can detect plumes, i.e., visible or measurable discharges of contaminants in water or in the air, and actors can reactively take countermeasures. Similarly, motion, acoustic, or light sensors in a building can detect the presence of intruders and command cameras or other instruments to track them. Alternatively, mobile actors can be moved to the area where the intruder has been detected to get high resolution images, prompt or block the intruder. Another promising application for WSANs are so-called pursuit evasion games (PEGs) [96], where a group of pursuer actors is required to detect, chase and capture a group of evaders in minimum time with the aid of a sensor network. As a last example, in earthquake scenarios sensors can help locate survivors and guide mobile actors performing rescue operations.

1.2 Previous Work

Although a few recent papers are specifically concerned with coordination and communication problems in sensor and actor networks, the literature on the subject is extremely limited. In [13], research challenges in wireless sensor and actor networks are discussed, and open research issues are described. In particular, several application scenarios are outlined, along with the existing challenges for effective sensor-actor coordination and actor-actor coordination.

In [106], the problem of "hazards" is considered, which consists of out-of-order execution of queries and commands due to a lack of coordination between sensors and actors. Three types of hazards are identified, and their undesirable consequences are shown. The authors also identify and enumerate the associated challenges in addressing hazards, and

propose a distributed hazard-free approach that addresses the problem and its challenges. The problem of avoiding hazards due to out-of-order execution of queries is of great importance in WSANs, and complementary to the proposed work.

In [107], the authors deal with the problem of mutual exclusion in WSANs. The problem of mutual exclusion consists if determining the minimum subset of actors that covers the entire event region such that there is minimal overlap in the acting regions. An example would be poison gas actors, where one dose of the gas merely invalidates subject, but two doses can kill. The problem is related, although different from, the actor-actor coordination problem, that will be discussed in Chapters 2 and 3.

A delay-energy aware routing protocol (DEAP) designed for sensor and actor networks is presented in [34], which enables a wide range of tradeoffs between delay and energy consumption, including an adaptive energy management scheme that controls the wake-up cycle of sensors based on the experienced packet delay. However, the paper only focuses on sensor-actor communication and assumes predetermined sensor-actor associations.

Some recent papers [50][37] have considered the issue of real-time communication in sensor networks, which is strictly related to the proposed research. The SPEED protocol [50] provides real-time communication services, and is designed to be a stateless, localized algorithm with low control overhead. End-to-end soft real-time communication is achieved by maintaining a desired delivery speed across the sensor network through a combination of feedback control and non-deterministic geographic forwarding. MMSPEED [37] is a significant extension of SPEED that can differentiate between flows with different delay and reliability requirements. MMSPEED is based on a cross-layer approach between the network and the MAC layers in which a judicious choice is made over reliability and timeliness of packet arrival. It is argued that the differentiation in reliability is an effective way of channeling resources from flows with relaxed requirements to flows with tighter requirements. Importantly, a new metric called *on-time reachability* is introduced, which is a measure of the probability that a packet reaches its destination within given delay

bounds. While current research directions make an effort to provide real-time streaming, they still provide best effort services. Giving firm delay guarantees in a dynamically changing network is a difficult problem and yet it is important for seamless real-time delivery of multimedia content. MMSPEED takes the step towards this end by adopting a probabilistic approach but clearly, further work is needed in this area.

Moreover, SPEED and MMSPEED try to provide real-time delivery of individual flows from different sensors. While this is required in several application scenarios, several sensor/actor applications may not need reliable and timely information on a flow basis. In line with this, the solutions proposed in Chapters 2 and 3 are based on a collective notion of reliability that is associated to the overall event and not to each individual flow. Besides, none of the above papers deals with sensor-actor coordination or with actor-actor coordination. In general, as discussed in [103], there are many open research challenges to enable real-time communication and coordination in sensor networks, especially due to resource constraints and scalability issues.

Several solutions propose to guarantee scalability and energy efficiency based on partitioning the sensor network into different clusters [51][111][61]. Most of the existing clustering algorithms can be classified as *topology dependent*, i.e., clusters are predetermined, depend on the topology of the sensor network, and may be adaptively reconfigured to deal with mobility or failure of the sensor nodes. Usually, in topology-dependent clustering one of the sensors is elected as a *clusterhead* in each cluster. When an event occurs, each sensor is already associated with a clusterhead. Conversely, similar to the dynamic clustering approach in [26], in the event-driven partitioning paradigm sensors are partitioned based on the characteristics of the event.

1.3 Organization of the Thesis

This thesis is organized as follows. In Chapter 2, coordination and communication problems in WSANs are jointly addressed in a unifying framework. A sensor-actor coordination

model is proposed, based on an *event-driven partitioning* paradigm. Sensors are partitioned into different sets and each set is constituted by a data-delivery tree associated with a different actor. The distributed partitioning is triggered by an event and data delivery trees are created *on-the-fly* to optimally react to the event itself and provide the required reliability with minimum energy expenditure. The optimal solution for the partitioning strategy is determined by mathematical programming, and a distributed solution is also proposed. In addition, a new model for actor-actor coordination is introduced. The actor coordination is formulated as a task assignment optimization problem for a class of coordination problems in which the area to be acted upon needs to be optimally split among different actors. An auction-based distributed solution of the problem is also presented.

Performance evaluation shows how global network objectives, such as compliance with real-time constraints and minimum energy consumption, can be achieved in the proposed framework with simple interactions between sensors and actors that are suitable for large-scale networks of energy-constrained devices.

In Chapter 3, coordination and communication problems in WSANs with mobile actors are studied. A hybrid location management scheme is introduced to handle the mobility of actors with minimal energy expenditure. First, a new location management scheme is proposed to handle the mobility of actors with minimal energy expenditure for the sensors, based on a hybrid strategy that includes *location updating* and *location prediction*. Actors broadcast location updates limiting their scope based on Voronoi diagrams, whereas sensors predict the movement of actors based on Kalman filtering of previously received updates. An optimal energy-aware forwarding rule is then derived for sensor-actor communication, based on geographical routing. The proposed scheme allows controlling the delay of the data-delivery process based on power control, and deals with network congestion by forcing multiple actors to be recipients for traffic generated in the event area. The motion of actors is coordinated to optimally accomplish the tasks based on the characteristics of the events.

In Chapter 4, we outline the research challenges for the delivery of multimedia content.

The state of the art in algorithms, protocols, and hardware for streaming of multimedia content in sensor and actor networks is discussed, and open research issues are discussed in detail. Architectures for WMSANs are explored, along with their advantages and drawbacks. Currently off-the-shelf hardware as well as available research prototypes for WMSANs are listed and classified. Existing solutions and open research issues at the application, transport, network, link, and physical layers of the communication stack are investigated, along with possible cross-layer synergies and optimizations.

In Chapter 5, we describe a new cross-layer communication architecture whose objective is to reliably and flexibly *deliver QoS to heterogeneous applications in WMSANs*, by carefully leveraging and controlling interactions among layers according to the applications requirements. Unlike the traditional networking paradigm, where functional separation among different layers has been shown to lead to modular and simple design, we show that by controlling interactions among functionalities handled at different layers of the protocol stack we can achieve the performance required in WMSANs, without sacrificing on modularity of the overall design.

Finally, in Chapter 6, we draw the main conclusions and outline future research directions.

CHAPTER II

COMMUNICATION AND COORDINATION IN WIRELESS SENSOR AND ACTOR NETWORKS

2.1 Preliminaries

As previously discussed, WSANs can be seen as a distributed control system designed to timely react to sensor information with an effective action. For this reason, *real-time coordination and communication* is an important concern in WSANs to guarantee timely execution of the right actions. The *energy efficiency* of network communications is also crucial, since sensors are resource-constrained devices with limited battery lifetime [15]. Furthermore, sensor network protocols and algorithms should be *scalable* and *localized*, as the number of nodes can be arbitrarily high.

For these reasons, in WSANs it is required to deliver event data in an energy-efficient way while abiding by real-time delay constraints, i.e., the operating point of the network should be cooperatively adjusted by sensors and actors to minimize the energy consumption, given some requirements on the acceptable delay. Another primary requirement of WSANs is their *scalability*, i.e., the property of performing operations efficiently when the number of network devices increases. For this reason, in this chapter we study the *sensor-actor coordination* based on a *Geographical Routing* paradigm [36]. The latter, also known as *Position Based Routing*, is becoming the most promising scalable solution for critically energy-constrained sensor networks, driven by the recent availability of small, inexpensive, and low-power GPS (Global Positioning System) receivers, along with techniques to distributively calculate relative sensor coordinates, e.g., based on signal strength [52]. Importantly, the nature of a sensor network usually imposes that each device be associated with its location to extract meaningful data from the environment. Hence, it becomes

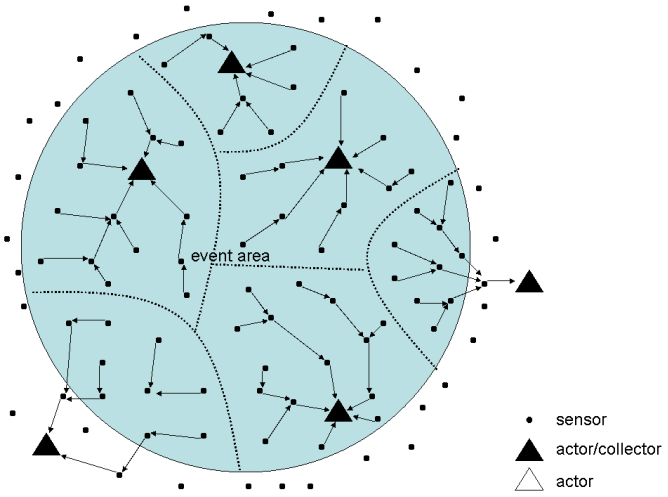


Figure 1: Event-driven partitioning with multiple actors.

natural to leverage localization information to perform the routing functionality as well.

Given the above requirements, we study communication and coordination problems in WSANs. In particular, we propose to base the sensor-actor coordination on an *event-driven partitioning* paradigm. Sensors are partitioned into different sets and sensors in each set construct a data-delivery tree associated with a different actor. The distributed partitioning is triggered by an event and data delivery trees are created *on-the-fly* to optimally react to the event itself and to provide the required reliability with minimum energy expenditure. This way, only sensors in the event area are partitioned, and each component of the partition consists of those sensor nodes that send their data to the same actor. Hence, event information is collected at the optimal actors while existing energy resources are better utilized, since sensors are partitioned based on location and scope of the event and on the position of actors. The resulting architecture is shown in Fig. 1. The event-driven partitioning approach also eliminates the communication overhead to maintain clusters before the event occurs, which is desirable especially in application scenarios where events are rare. Our approach is in line with the *dynamic clustering* mechanism in [26][57]. However, assumptions in [26][57] are strictly tied to the particular application considered, i.e., target tracking of moving objects, while we consider a more general framework. In addition,

we introduce a model for actor-actor coordination, whose objective is to optimally allocate tasks to different actors to collaboratively achieve a global goal. We define an optimization model that describes a class of coordination problems in which the area to be acted upon needs to be optimally split among different actors, depending on the actor capabilities.

This chapter is the first to propose solutions tailored for WSANs, and introduces a framework for communication and coordination problems in WSANs, whose overall contribution can be outlined as:

Sensor-actor coordination:

- We propose an optimization model based on the event-driven partitioning paradigm for sensor-actor coordination. This defines how sensors communicate with actors, which actor is fed by each sensor, and which data paths should be established between sensors and actors. Furthermore, we propose a new notion of *reliability* that accounts for the timely delivery of data packets at the network layer. The objective is to comply with the reliability requirement of the application and minimize the energy consumption. We determine the optimal strategy for event-driven partitioning by *Integer Linear Programming (ILP)* [11].

- We propose a multi-state distributed algorithm that determines sensor-actor data paths and implicitly partitions sensors in the event area as the event occurs. The algorithm achieves an energy-efficient solution for sensor-actor coordination and is based on an adaptive mechanism that trades off energy consumption for delay when event data must be delivered to the actors within pre-determined latency bounds.

Actor-actor coordination:

- We define the actor-actor coordination problem as a task assignment problem and propose a solution for a class of coordination problems in which the area to be acted upon needs to be optimally split among different actors. The action workload is thus

divided among different, possibly heterogeneous, actors, depending on the characteristics of the event. The task assignment problem is formulated as a *Mixed Integer Non-Linear Program (MINLP)* [31].

- We propose a localized distributed solution for the actor-actor coordination problem, based on an analogy with an auction mechanism among the actors.

Since WSANs can enable a broad range of applications with different requirements, we focus on scenarios with immobile actors that can act on a limited area defined by their *action range*[1], and the area where the event occurs needs to be monitored for a prolonged period of time. Our ultimate objective is to demonstrate how global network objectives, such as compliance with real-time constraints and minimum energy consumption, can be reached in the proposed framework with simple interactions between sensors and actors, suitable for large-scale networks of energy-constrained devices. As a representative application, consider a system of simple scalar sensors that collaboratively detect the presence of an intruder. Lower tier sensors could wake-up on demand pan-tilt-zoom camera/actors mounted on robotic arms that take images and video streams from the areas where the event has been detected based on sensor input. The coverage of a pan-tilt-zoom camera is defined by its field of regard, i.e., the points of the physical environment that can be perceived by the camera given its ability to reposition. This would correspond to the action area of the camera. Cameras whose field of regard are overlapped would collaboratively decide on which camera-actor is best suited to gather images from the area based on the proposed model.

The remainder of this chapter is organized as follows. In Section 2.2 we state the sensor-actor coordination problem and propose an integer linear programming formulation, while in Section 2.3 we propose a distributed solution. In Section 2.4, we state the actor-actor

[1]Note that actors are immobile from a networking perspective, but may contain moving mechanical parts. The notion of action range may refer for example to the extension of mechanical arms that perform an action, or to the range of automatic water sprinklers.

coordination problem, while in Section 2.5 we introduce a distributed solution based on a real-time localized auction mechanism. Detailed comparative performance evaluation and simulation results are presented in Section 2.6.

2.2 Sensor-Actor Coordination: Problem Formulation

As discussed in the previous section, sensor-actor communications may have real-time requirements. Hence, we introduce a novel notion of reliability that accounts for the percentage of packets generated by the sensors in the event area that are received within a pre-defined latency bound (which we refer to as *reliable packets*). Unlike other notions of reliability, the definition introduced here is related to the real-time delivery of data packets from sources to actors, and is calculated at the network layer.

Definition 1 *The* latency bound B *is the maximum allowed time between the instant when the physical features of the event are sampled by the sensors and the instant when the actor receives a data packet describing these event features.*

Definition 2 *A data packet that does not meet the latency bound B when it is received by an actor is said to be* expired *and thus,* unreliable. *Similarly, a data packet received within the latency bound B is said to be* unexpired *and thus,* reliable.

Definition 3 *The* event reliability r *is the ratio of* reliable *data packets over all the packets generated in a decision interval*[2]. *The* event reliability threshold r_{th} *is the minimum* event reliability *required by the application.*

Definition 4 *The* lack of reliability *is the difference $(r_{th} - r)$ between the required event reliability threshold r_{th} and the observed event reliability r at a given time. A negative lack of reliability indicates a reliability above the required threshold and is also referred to as an* excess of reliability.

[2] *Whenever one or more packets are dropped by an intermediate sensor, the actor is notified about the lost packet(s) in the header of the next data packet, so that the packet loss can be taken into account in the computation of the reliability.*

Note that the latency bound B and the event reliability threshold r_{th} are dependent on the application requirements.

The sensor-actor coordination problem consists of establishing data paths from each sensor residing in the event area to the actors by i) ensuring that the observed reliability r is above the event reliability threshold r_{th} (i.e., $r \geq r_{th}$); ii) minimizing the energy consumption associated with data delivery paths.

We refer to our solution for the sensor-actor coordination problem as *event-driven partitioning with multiple actors* and model it as an Integer Linear Program (ILP). In Section 2.2.1 we describe the network and energy model. In Section 2.2.2 we provide the complete ILP formulation of the problem.

2.2.1 Network and Energy Model

The network of sensors and actors is represented as a graph $\mathcal{G}(\mathcal{S}^{\mathcal{V}}, \mathcal{S}^{\mathcal{E}})$, where $\mathcal{S}^{\mathcal{V}} = \{v_1, v_2, \ldots, v_N\}$ is a finite set of nodes (vertexes) in a finite-dimension terrain, with $N = |\mathcal{S}^{\mathcal{V}}|$, and $\mathcal{S}^{\mathcal{E}}$ is the set of links (edges) among nodes, i.e., $e_{ij} \in \mathcal{S}^{\mathcal{E}}$ iff nodes v_i and v_j (also i and j for simplicity in the following) are within each other's transmission range. Let $\mathcal{S}^{\mathcal{A}}$ represent the set of actors, with $N_A = |\mathcal{S}^{\mathcal{A}}|$. We refer to an actor that is collecting traffic from one or more sources as *collector*. Let $\mathcal{S}^{\mathcal{S}}$ be the set of traffic sources, with $N_S = |\mathcal{S}^{\mathcal{S}}|$. This set represents the sensor nodes that detect the event, i.e., the sensors that reside in the event area. Since the set of sources is disjoint from the set of actors, $\mathcal{S}^{\mathcal{A}} \subset \mathcal{S}^{\mathcal{V}}$, $\mathcal{S}^{\mathcal{S}} \subset \mathcal{S}^{\mathcal{V}}$, and $\mathcal{S}^{\mathcal{A}} \cap \mathcal{S}^{\mathcal{S}} = \emptyset$. We define $\mathcal{P} = \{(s, a) : s \in \mathcal{S}, a \in \mathcal{A}\}$ as the set of source-destination connections. Although the information rate of each connection can be described by the traffic matrix $P = [p_{sa}]$, where p_{sa} represents the average information rate (bits/s) between a source node $s \in \mathcal{S}$ and an actor node $a \in \mathcal{A}$, we shall assume that all sources generate 1 bit of information. Hence, we can omit the traffic matrix from the model as the solution of the problem does not depend on the traffic matrix when all sources generate information at the same data rate, as is the case in many applications of interest.

An accurate model for energy consumption per bit at the physical layer is $E = E_{elec}^{trans} + \beta d^{\alpha} + E_{elec}^{rec}$, where E_{elec}^{trans} is a *distance-independent* term that takes into account overheads of transmitter electronics (PLLs, VCOs, bias currents, etc.) and digital processing; E_{elec}^{rec} is a *distance-independent* term that takes into account the overhead of receiver electronics, while βd^{α} is a *distance-dependent* term that accounts for the radiated power necessary to transmit one bit over a distance d between source and destination. As in [51], we assume that $E_{elec}^{trans} = E_{elec}^{rec} = E_{elec}$. Thus, the overall expression simplifies to $E = 2E_{elec} + \beta d^{\alpha}$, where α is the exponent of the path loss ($2 \leq \alpha \leq 5$), β is a constant $[J/(bits \cdot m^{\alpha})]$, and E_{elec} is the energy needed by the transceiver circuitry to transmit or receive one bit $[J/bits]$.

In our energy model we consider that, when a sensor node receives data from at least two other nodes, it aggregates the received information by *data fusion* [15], i.e., a single packet is created by merging multiple incoming packets, thus reducing the amount of data to be transmitted. To effectively support this function, an algorithm for data fusion should be implemented on each sensor, which is out of the scope of this chapter. Moreover, we ignore the processing cost in our model, since the processing cost is much lower than the communication cost. This is justified by experimental results on sensor network prototypes such as [85], where the energy necessary to transmit 1 kbit is shown to be equivalent to the energy necessary to execute 300,000 processor instructions.

2.2.2 Integer Linear Program

The objective of the optimization problem is to find *data aggregation trees* (da-trees) from all the sensors that reside in the event area (referred to as sources) to the appropriate actors. A da-tree is composed by aggregating individual *flows*, where a flow is defined as a connection between a sensor and an actor. All leaves in a da-tree are sources (but not all sources are necessarily leaves), and each actor is either the root of a da-tree or does not participate in the communication. The da-trees are constructed in such a way that each source belongs to one tree only, and each tree has only one actor as its root. Therefore, each source is

associated with an actor to achieve an optimal event-driven partition. In fact, event-driven partitioning can be seen as a joint twofold problem: i) select the optimal subset of actors to which sensor readings will be transmitted; ii) construct the minimum energy da-trees towards those selected actors that meet the required event reliability constraint. The union of all trees rooted at the actors implicitly partitions the set of source nodes in the event area. Figure 1 gives an example of this configuration.

The optimal strategy for event-driven partitioning is formulated as an *Integer Linear Program* (ILP) [11]. The network topology is assumed to be *1-connected*, i.e., at least one path exists between each sensor and actor. Note that this is not a strict requirement in dense sensor networks. We introduce the following notations that are used in the problem formulation:

- e_{ij} is a binary variable representing a link, that equals 1 iff nodes i and j are within each other's transmission range;
- c_{ij} is the cost of the link between nodes i and j, i.e., $2E_{elec} + \beta d_{ij}^{\alpha}$, where d_{ij} is the distance between nodes i and j;
- x_{ij}^{k} is a binary variable that equals 1 iff link (i, j) is part of the da-tree associated with actor k;
- $f_{ij}^{k,s}$ is a binary variable that equals 1 iff source s sends data to actor k and link (i, j) is in the path from s to k;
- $l^{k,s}$ is a binary variable that equals 1 iff sensor s sends data to actor k;
- p_{ij} is the propagation delay associated with link (i, j), defined as d_{ij}/v, where v is the signal propagation speed;
- $\tilde{d}$ is a parameter that accounts for *processing*, *queuing*, and *medium access* delay at each sensor node;

- B is the latency bound on each source-actor flow;
- r and r_{th} are the event reliability and the required event reliability threshold, respectively;
- $b^{k,s}$ is a binary variable that equals 1 iff the connection between source s and actor k is not compliant with the latency bound, i.e., the end-to-end delay is higher than the latency bound B;
- Q is the number of non-compliant sources.

The problem can be cast as follows:

$\mathbf{P}_{Min}^{Com}$: Event-driven Partitioning with Multiple Actors

$$
\begin{aligned}
Given: &\quad e_{ij},\ c_{ij},\ p_{ij},\ v,\ \tilde{d},\ B,\ r_{th} \\
Find: &\quad x_{ij}^{k},\ f_{ij}^{k,s},\ l^{k,s},\ b^{k,s},\ r \\
Minimize: &\quad C^{TOT} = \textstyle\sum_{k\in\mathcal{S}^{\mathcal{A}}} \sum_{(i,j)\in\mathcal{S}^{\mathcal{E}}} x_{ij}^{k}\cdot c_{ij} + \gamma\cdot Q \qquad (1) \\
Subject\ to: &
\end{aligned}
$$

$$\sum_{j\in\mathcal{S}^{\mathcal{V}}} (f_{sj}^{k,s} - f_{js}^{k,s}) = l^{k,s}, \forall s\in\mathcal{S}^{\mathcal{S}}, \forall k\in\mathcal{S}^{\mathcal{A}}; \qquad (2)$$

$$\sum_{j\in\mathcal{S}^{\mathcal{V}}} (f_{kj}^{k,s} - f_{jk}^{k,s}) = -l^{k,s}, \forall s\in\mathcal{S}^{\mathcal{S}}, \forall k\in\mathcal{S}^{\mathcal{A}}; \qquad (3)$$

$$\sum_{j\in\mathcal{S}^{\mathcal{V}}} (f_{ij}^{k,s} - f_{ji}^{k,s}) = 0,$$

$$\forall s\in\mathcal{S}^{\mathcal{S}}, \forall k\in\mathcal{S}^{\mathcal{A}}, \forall i\in\mathcal{S}^{\mathcal{V}}\ s.t.\ i\neq s, i\neq k; \qquad (4)$$

$$f_{ij}^{k,s} \leq e_{ij}, \forall s\in\mathcal{S}^{\mathcal{S}}, \forall k\in\mathcal{S}^{\mathcal{A}}, \forall i\in\mathcal{S}^{\mathcal{V}}, \forall j\in\mathcal{S}^{\mathcal{V}}; \qquad (5)$$

$$f_{ij}^{k,s} \leq x_{ij}^{k}, \forall s\in\mathcal{S}^{\mathcal{S}}, \forall k\in\mathcal{S}^{\mathcal{A}}, \forall i\in\mathcal{S}^{\mathcal{V}}, \forall j\in\mathcal{S}^{\mathcal{V}}; \qquad (6)$$

$$\sum_{k\in\mathcal{S}^{\mathcal{A}}} l^{k,s} = 1, \forall s\in\mathcal{S}^{\mathcal{S}}; \qquad (7)$$

$$f_{ij}^{k,s} \leq l^{k,s}, \forall s \in \mathcal{S}^{\mathcal{S}}, \forall k \in \mathcal{S}^{\mathcal{A}}, \forall i \in \mathcal{S}^{\mathcal{V}}, \forall j \in \mathcal{S}^{\mathcal{V}}; \tag{8}$$

$$\varepsilon \cdot [B - \sum_{(i,j) \in \mathcal{S}^{\mathcal{E}}} f_{ij}^{k,s}(p_{ij} + \tilde{d})] \leq b^{k,s}, \ \ \forall s \in \mathcal{S}^{\mathcal{S}}, \forall k \in \mathcal{S}^{\mathcal{A}}; \tag{9}$$

$$Q = \sum_{k \in \mathcal{S}^{\mathcal{A}}} \sum_{s \in \mathcal{S}^{\mathcal{S}}} b^{k,s}; \qquad r = \frac{|\mathcal{S}^{\mathcal{S}}| - Q}{|\mathcal{S}^{\mathcal{S}}|} \geq r_{th}. \tag{10}$$

The objective function in (1) minimizes the overall energy consumption and imposes a penalty by multiplying the number Q of non-compliant sources by a penalty coefficient γ whose value must be high enough (e.g., orders of magnitude higher than the energy consumption) to guarantee uniqueness of the solution. This allows minimizing the number of non-compliant sources Q in (10) with a single-objective problem. As previously discussed, a flow is a connection between a source and an actor. Flows associated with the same actor are aggregated in a da-tree. Constraints (2), (3), and (4) express conservation of flows [11], i.e., each source generates a flow, which is collected by an actor. In particular, constraint (2) guarantees that a source node generates a flow on the tree of the selected actor, and only on that one; while non-source nodes do not generate any flow. Constraint (3) requires that flows generated by each source be collected by one actor only. Constraint (4) imposes that the balance between incoming and outgoing flows is null for non-source and non-actor nodes. Constraint (5) ensures that flows are created on links between adjacent nodes (i.e., that are within transmission range of each other). Constraint (6) forces all flows from different sources but directed towards the same actor to be aggregated in the tree associated with that actor. Constraint (7) imposes that each source send data to exactly one actor. Constraint (8) ensures that all flow variables from a source to a particular actor are zero unless that actor is selected by the source. Constraint (9) requires that the binary variable $b^{k,s}$ be equal to 1 if and only if the flow between source s and actor k violates the latency bound B. The small negative coefficient ε scales the value in the square parentheses to make it smaller than 1. Hence, when the latency bound is violated, the left side of (9) is a small positive value, which forces the binary variable $b^{k,s}$ to be 1. Conversely, when the latency bound is met, the left side of (9) is negative and $b^{k,s}$ will assume the 0 value to minimize

the objective function in (1). Finally, in (10), Q is defined as the number of non-compliant sources and the reliability r is calculated as the ratio of compliant sources over all sources and constrained to be over the required threshold.

Since $\mathbf{P}_{\mathbf{Min}}^{\mathbf{Com}}$ is ILP, it can be shown [41] that it is at least as complex as the Geometric Connected Dominating Set problem, which is proven to be NP-complete. Hence, $\mathbf{P}_{Min}^{Com}$ is NP-complete. However, it is still possible to solve $\mathbf{P}_{Min}^{Com}$ for networks of moderate size (up to 100 nodes), as will be shown in Section 2.6. This allows gaining insights on the properties of the optimal solution, and designing distributed solutions that try to reproduce characteristics of the optimal network configuration. The design of the distributed protocol presented in the following section is based upon the analysis and performance study of the above problem. This includes the observation than nodes tend to send data towards the closest actor to minimize the energy consumption, unless another node allows performing aggregation at a lower cost. Moreover, the mathematical formulation constitutes a fundamental benchmark for the energy consumption of any possible distributed solution. In this spirit, in Section 2.6.1 we will use it as a benchmark for the performance of the distributed suboptimal but scalable algorithm introduced in the following section.

2.3 Sensor-Actor Coordination: Distributed Protocol

The objective of the distributed protocol proposed in this section is to build da-trees between the sources that reside in the event area and the actors in such a way as to minimize the objective function in (1), i.e., to provide the required reliability r_{th} with minimum energy expenditure. As will be shown in Section 2.6.1, the proposed protocol constructs da-trees between sources and actors that can be seen as an approximate solution to the event-driven partitioning with multiple actors problem, described in Section 2.2, and was designed to reproduce as closely as possible performance results yielded by that formulation. We refer to the protocol as *Distributed Event-driven Partitioning and Routing* (DEPR)

protocol. The performance of this distributed protocol will be compared to the optimal solution in Section 2.6.

As discussed in [74], in geographical routing algorithms *localized routing decisions*, i.e., based on local topology information, can lead to data paths whose energy efficiency is close to the global optimum. This means that, in densely deployed sensor networks, topology information related to network regions that are "far" from where the routing decision is being taken are not essential. For this reason, the objective of the proposed protocol is to minimize the energy consumption by relying on local information and on *greedy* routing decisions. This way, we intuitively seek a solution that results in a good compromise between the energy efficiency of the established da-trees, and the amount of topology information needed by each sensor to take a routing decision [74]. Conversely, complying with pre-determined delay bounds requires some form of end-to-end feedback. Instead of requiring feedback information for each individual source, which would cause unacceptable overhead, we rely on collective feedback from the receiving actors, as will be explained in Section 2.3.5. Each actor advertises the observed reliability. Based on this, the proposed protocol favors local behavior for each individual sensor node that results in a global network behavior that is compliant with the application requirements, i.e., provide event reliability r above the required threshold r_{th} (Definition 3 in Section 2.2) and minimize the energy consumption. The reliability is controlled based on the idea of adjusting the delays, by modifying the average end-to-end path length. While modifying the energy consumption in an ad hoc network by changing the transmitted power is common practice, the proposed protocol can be also seen as a mechanism to adjust the end-to-end delay based on transmission power control. To the best of our knowledge, this idea has not been thoroughly explored so far.

In the description of the DEPR protocol, we assume that: i) each sensor is aware of its position, as the sensor node can be equipped with a GPS receiver or the position can be determined by means of localization techniques [52]; ii) each sensor is aware of the position

of its neighbors, as every node locally broadcasts its position; and of the actors, as each actor periodically beacons its position in the sensor field; iii) the network is synchronized by means of one of the existing synchronization protocols [105]. A study on the impact of localization and synchronization errors is left for future work.

An important issue in geographical routing algorithms is to avoid routing loops. Hence, we introduce some concepts related to *path loop freedom*.

Definition 5 *Given nodes v and x, the* absolute advance *of node x, with respect to v, is the distance between v and its closest actor c_v minus the distance between x and its closest actor c_x*[3].

Definition 6 *Given nodes v and x, the* advance towards the collector c *of x, with respect to v, is the distance between v and c minus the distance between x and c.*

Intuitively, if x has *positive absolute advance* with respect to v, this means that x is closer to one (whatever) actor than v. If x has *positive advance towards collector* c with respect to v, x is closer to actor c than v. For any multi-hop path, a *positive absolute advance* at every hop guarantees loop freedom, irrespective of the final destination, since at each hop the packet is closer to a collector than at the previous hop. A *positive advance towards an actor* c at every hop guarantees a loop-free path from a source node to the actor c.

2.3.1 Overview of DEPR

The objective of the proposed protocol is to create da-trees between the sources and a subset of the actors, referred to as collectors. A da-tree is thus created between each collector and the sources associated with that collector. This way, the set of sources is implicitly partitioned, with each part composed of the sources associated with a single collector.

[3] *Note that c_v and c_x can be different actors.*

Each sensor alternates among four different states, namely *idle*, *start-up*, *speed-up*, and *aggregation state*, plus an additional *recovery state* that will be discussed in Section 2.3.6. An overview of the state transitions is depicted in Fig. 2. The main objective of these state transitions is to reduce the number of hops, which results in decreased delay, when the reliability requirement is violated; and to save energy when the reliability requirement is met. This is achieved by probabilistically modifying the behavior of the sensor nodes at the routing layer and physical layer, i.e., inducing them to select their next hop so as to increase the delay and reduce the energy consumption when the reliability is high, and viceversa reducing the delay at the expense of energy consumption when the reliability is low. This is achieved by dynamically adjusting the transmit power at the same time.

In multi-hop wireless networks, transmit power control is a typical action with cross-layer implications [60]. The transmit power level affects the quality of the received signal and thus impacts packet error rate and energy consumption at the physical layer; it determines the communication range and thus affects the pool of feasible next hops at the network layer; it impacts the magnitude of the interference that affects the capacity of the channel, thus influencing congestion and affecting the transport layer. Therefore, power control impacts several network performance measures such as throughput, delay and energy consumption. Specifically, choosing a high transmission power, coupled with an appropriate network layer policy, may reduce the number of hops needed to reach the intended destination, while increasing interference and exposed terminals in the shared wireless medium. A high transmission power also improves the connectivity of the network by increasing the number of direct links seen by each node but this is at the expense of reducing spatial re-use. As will be shown later in our simulations, in case of low or moderate traffic reducing the end-to-end path lengths leads to reduced end-to-end delay and increased energy consumption. On the other hand, choosing a lower transmission power reduces the interference seen by potential transmitters and to a certain extent reduces the energy consumption and thus, increases the network lifetime [48]. However, a low transmission power requires

more forwarding nodes resulting in higher end-to-end latency and increases the number of hidden terminals. Therefore, an efficient power control scheme should determine how to trade off the multiple objectives of network capacity, battery life and latency. This is also the objective of our adaptive distributed protocol.

Source nodes add a timestamp to the event data packet that they transmit to the actors, to allow the corresponding actors to compute the delay of each packet. For each decision interval, each actor then computes the event reliability r as the ratio of unexpired packets over all generated packets and periodically broadcasts its value. Sensor nodes associated with that collector base their state transitions on the reliability observed by the collector, which is broadcast at the end of each decision interval. When the advertised value r is below the so-called *low event reliability threshold* r_{th}^{-}, where $r_{th}^{-} = r_{th} - \epsilon^{-}$, i.e., the lack of reliability ($r_{th} - r$) is above a certain positive margin ϵ^{-}, then it is necessary to speed-up the data delivery process by reducing the end-to-end delay. Conversely, when the advertised value r is above the so-called *high event reliability threshold* r_{th}^{+}, where $r_{th}^{+} = r_{th} + \epsilon^{+}$, i.e., the excess of reliability ($r - r_{th}$) is above a certain margin ϵ^{+}, then there is reliability in excess that can be traded off for energy savings. The coefficients ϵ^{+} and ϵ^{-} are needed to define a "tolerance zone" around the required reliability threshold for practical purposes (i.e., reduce oscillations). Good values for ϵ^{+} and ϵ^{-} find a good compromise between stability and tolerance, and can be determined by simulation, as discussed in Section 2.6.2.

Each sensor node starts in an idle state, where it samples the environment and monitors the channel for incoming data packets. A sensor enters the start-up state when it either senses an event or it receives the first data packet from a neighboring sensor. The collective operation of the sensor nodes in the start-up state allows to timely establish paths to an actor for each source that resides in the event area, as described in detail in Section 2.3.2. These paths constitute a good compromise between latency and energy consumption. To distributively achieve this objective, each sensor node in the event area enters the start-up state, and starts sending data packets towards an actor.

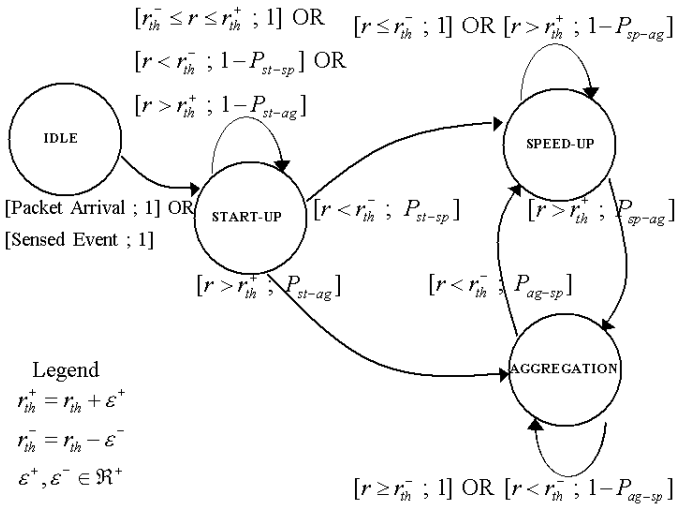


Figure 2: State transition diagram for a sensor node.

Sensor nodes expect feedback messages from the collector/actor they are associated with. If the event reliability r is advertised to be below the low event reliability threshold r_{th}^-, it is necessary to reduce the sensor-actor delay, by reducing the end-to-end path length. Hence, when it receives a packet advertising a reliability below the low reliability threshold ($r < r_{th}^-$), a sensor in the start-up state enters the speed-up state with probability P_{st-sp}, which can be a monotonically increasing function of the *lack of reliability*. This is further discussed in Section 2.6.2. The notation $[cond; P]$ in Fig. 2 indicates a transition that occurs with probability P when condition $cond$ is verified.

If the event reliability r is above the high event reliability threshold r_{th}^+ (i.e., $r > r_{th}^+$), it is possible to save energy. In this case, a node in the start-up state enters the aggregation state with probability P_{st-ag}, which can be a monotonically increasing function of the *excess of reliability*, where it tries to minimize the energy consumption associated with its transmission by relaying data to the closest neighbor that participates in a da-tree.

Then, sensors can alternate between the speed-up and the aggregation state in order to respond to feedback messages from collectors. Hence, as shown in Fig. 2, a sensor in the speed-up state enters the aggregation state with probability P_{sp-ag} when $r > r_{th}^+$, while a sensor in the aggregation state enters the speed-up state with probability P_{ag-sp}

when $r < r_{th}^{-}$. P_{ag-sp} may increase with increasing advertised lack of reliability, while P_{sp-ag} may increase with increasing excess of reliability. The objective of the protocol is to converge to a solution with reliability close to the event reliability threshold with minimal energy consumption. A sensor goes back to the idle state if it does not generate or receive packets for $idleTimeout$ seconds.

Note that a probabilistic policy prevents system oscillations that would occur if all sensors changed state at the same time. Clearly, the proposed algorithm is particularly effective when the time needed to set-up and configure the tree is small as compared to the length of the subsequent monitoring and acting phase.

In the following we describe the operations of each state.

2.3.2 Start-up State

As shown in Fig. 2, a node enters the start-up state from the idle state when it detects an event, or when it receives a packet to be relayed to an actor. Sensor i in the start-up

Algorithm 1 Start-up State

Pseudo-code executed by node v_i in the start-up state
$mincost = \infty$
if (($I\_am\_a\_source$) **or** ($I\_am\_a\_relayer$)) **then**
  for each of my neighbors v_j **do**
    for each actor s_k **do**
      if ($2E_{elec} + \beta d_{ij}^{\alpha} + 2E_{elec} + \beta d_{js_k}^{\alpha}$) $< mincost$ **then**
        $mincost = 2E_{elec} + \beta d_{ij}^{\alpha} + 2E_{elec} + \beta d_{js_k}^{\alpha}$
        $nexthop$=v_j
      end if
    end for
  end for
end if
Inform $nexthop$ that it is a relayer

state, either as a source or as a relayer for a data packet, selects its next hop based on the so-called *two-hop rule*. According to the two-hop rule, node i selects as next hop among its neighbors the node j that minimizes the sum of the energy consumption from i to j and the energy consumption from j to the actor c_j closest to j, which is computed according

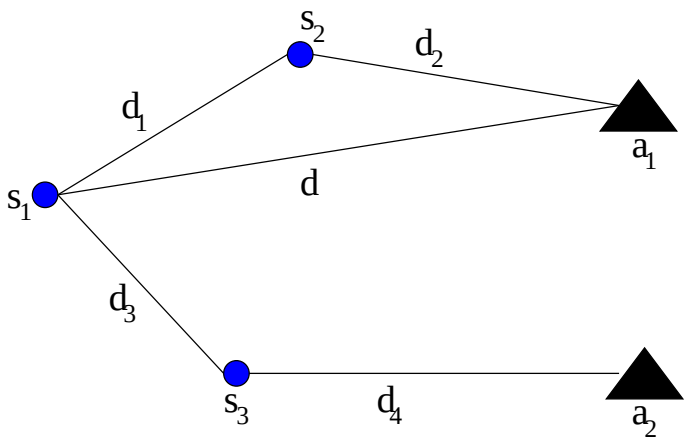


Figure 3: Loop freedom property of the two-hop rule.

to the link energy model introduced in Section 2.2.1. Hence, the energy consumption E_j associated with a neighbor j of i is

$$E_j = 2E_{elec} + \beta d_{ij}^{\alpha} + 2E_{elec} + \beta d_{jc_j}^{\alpha}, \tag{11}$$

where d_{ij} represents the distance between i and j, while d_{jc_j} represents the distance between j and its closest actor c_j. Note that the latter link may or may not exist. The energy metric on this link is considered to account for the advancement towards the destination of a particular neighbor. The two-hop rule selects as next hop the node j associated with the minimum two-hop energy consumption. As a result, the source-actor path will be established by applying the two-hop rule iteratively. Note that this procedure is only based on local position information. It requires each node to know only the position of its neighbors and of the actors, and does not entail any other exchange of information. Noticeably, since at each step the routing decision is independent of previous decisions, a packet intended for a certain actor by its generating source can be transmitted towards another destination actor by an intermediate node in the end-to-end path. For this reason, the collector actor transmits its identifier on the reverse da-tree in order to inform the source sensors about its own identity. The operations executed by a sensor node in the start-up state are detailed in Algorithm 1.

The two-hop rule produces loop-free paths, as stated below.

Lemma 1 *A next hop j selected by a node i with the* two-hop rule *has a* positive absolute advance *with respect to i (see Def. 5).*

We refer to Fig. 3. Assume node s_1 is holding a message that needs to be relayed to an actor, either a_1 or a_2. Let us also assume, without loss of generality, that the two-hop path s_1-s_2-a_1 is more energy efficient than the direct link s_1-a_1, i.e.,

$$4E_{elec} + \beta(d_1^\alpha + d_2^\alpha) \leq 2E_{elec} + \beta d^\alpha, \tag{12}$$

which leads to

$$(d_1^\alpha + d_2^\alpha) < d^\alpha. \tag{13}$$

Let us now assume that a_1 is the closest actor to node s_1 and that s_3 is the best next hop according to the two-hop rule, i.e., the energy necessary to reach a_2 through s_3 is lower than the energy required to reach a_1 through s_2, according to the energy metric in Section 2.2.1. This directly translates into

$$d_3^\alpha + d_4^\alpha < d_1^\alpha + d_2^\alpha. \tag{14}$$

Hence,

$$d_3^\alpha + d_4^\alpha < d_1^\alpha + d_2^\alpha < d^\alpha \tag{15}$$

which ultimately means, being $\alpha \geq 2$, that $d > d_4$. Therefore, the energy efficient next hop always has a positive absolute advance.

As a consequence, applying the two-hop rule at each hop produces a loop-free path between source and actor.

2.3.3 Speed-up State

The objective of the collaborative operation of nodes in the speed-up state is to minimize the number of hops between sources and actors. This is achieved by applying the Greedy Routing Scheme (GRS) [39] forwarding rule. According to GRS, each node sends the

packet to the node closest to the destination within the transmission range. It is intuitive that this rule minimizes the number of hops in the path, the distance traveled by the packet, and the number of transmissions of the same data packet. The pseudo-code of the operations executed by a sensor node in the speed-up state is reported in Algorithm 2. The set P_i in the algorithm represents the subset of neighbors of v_i with absolute positive advance with respect to v_i

Algorithm 2 Speed-up State

Pseudo-code executed by node v_i in the speed-up state
for each node $v_j \in P_i$ **do**
  if (distance(v_i,v_j)> distance(v_i,$next\_hop$)) **then**
    $next\_hop = v_j$
  end if
end for

2.3.4 Aggregation State

The objective of the aggregation state is to reduce the overall energy consumption. To this end, sensor nodes in the aggregation state take routing decisions that reduce the global energy consumption, by relying on the data fusion algorithm that we assume to be implemented on each sensor. Since data packets can be aggregated by any node in the network, the objective of a node in the aggregation state is to route data to the closest node in its neighborhood that is part of da-tree.

As previously discussed, after da-trees are established, each sensor knows which collector-actor it is associated with. By overhearing transmissions on the shared medium, each sensor learns which da-tree are its neighbors part of (if any, as some neighbor sensors may not even be in the event area), i.e., which collector actor are they associated with. Hence, node v_i in the aggregation state first evaluates the cost of transmitting data to those among its neighbors that are part of a da-tree. This way, it can identify a minimum-cost neighbor, i.e., the neighbor v_{min} that requires minimum energy consumption to be reached among those associated with one of the da-trees. We emphasize that this does not incur any overhead,

other than that caused by overhearing packets. Two different situations can occur. Node v_{min} can either be on the same da-tree as v_i, and hence associated with the same collector; or it can be in a different da-tree.

If v_{min} is in the same da-tree as v_i, v_{min} can be selected as next hop by v_i only if it has a positive advance towards the collector that both nodes are associated with, i.e., if v_{min} is closer than v_i to the collector (see Def. 6). This guarantees loop freedom. In the resulting da-tree, every parent node is guaranteed to have positive advance towards the collector, with respect to each child. When v_{min} is selected, the individual transmission cost for v_i is locally minimized and the overall cost of the tree is thus reduced. If v_{min} has no positive advance towards the collector with respect to v_i, v_i deletes v_{min} from the list of possible next hops and determines a new v_{min} among the remaining neighbors.

The other possible situation occurs when v_{min} is associated with a different collector than v_i, i.e., v_i and v_{min} are in two different da-trees. In this case, v_i is allowed to select v_{min} as its next hop only if v_i is a leaf in its da-tree and v_{min} has a positive advance towards its actor with respect to v_i. This guarantees loop-freedom of the overall tree, as every parent node is assured to have a positive advance towards the actor with respect to each child. Conversely, it can be easily shown that if non-leaf nodes are allowed to switch from one da-tree to another, loops may be created, as the condition that every parent node is closer to the actor than each child does not necessarily hold. The detailed operations executed by a sensor node in the aggregation state are reported in Algorithm 3.

2.3.5 State Transitions

The transitions of sensor nodes among the different states are driven by feedback messages from actors. Hence, the proposed mechanism can be considered as a form of closed-loop control at the network layer. Feedback messages are periodically sent by each actor, with period equal to Δf seconds. At each decision instant k, the actor feedback is determined

Algorithm 3 Aggregation State

Pseudo-code executed by a node v_i in the aggregation state
for each of my active neighbors v_j **do**
  if (distance(v_i,v_j) $<$ distance(v_i,$nexthop$)) **then**
    v_{min}=v_j
  end if
end for
s=actor(v_{min})
if ($s == myactor$) **then**
  if distance(v_{min},s)<distance(v_i,s) **then**
    $nexthop$=v_{min}
  else
    delete v_{min} from list and restart Aggregation State
  end if
else if $I\_am\_a\_leaf$ **then**
  $nexthop$=v_{min}
else
  delete v_{min} from list and restart Aggregation State
end if

based on three different reliability measures, namely the *reliability* $r[k]$, the *short-term reliability* $r_{sh}[k]$, and the *predicted reliability* $\hat{r}[k+1]$. The actor calculates the reliability $r[k]$ observed during the last decision interval, whose length is Δd, as discussed in Section **??**. Similarly, it calculates the so-called short-term reliability $r_{sh}[k]$, as the reliability observed during the last short decision interval, of length Δd_{sh}, with $\Delta d_{sh} < \Delta d$. Based on the current reliability $r[k]$ and on the history of past measurements $r[k-1]$, $r[k-2]$,..., the actor calculates the predicted reliability $\hat{r}[k+1] = f(r[k], r[k-1], r[k-2], ...)$. The feedback packet contains the advertised value of reliability $r_{adv}[k]$, calculated on the basis of these three measures, and is actually sent only if the advertised value of reliability is above the high reliability threshold r_{th}^{+} or below the low reliability threshold r_{th}^{-}. If the sensors receive no feedback, they assume that the reliability lies within r_{th}^{+} and r_{th}^{-}.

The operation of the adaptive control scheme run at each actor is summarized in Algorithm 4. The advertised reliability $r_{adv}[k]$ is calculated as follows. As a general rule, the advertised value $r_{adv}[k]$ at instant k is the predicted reliability $\hat{r}[k+1]$. This way, the actor tries to identify ongoing trends in the value of reliability and react accordingly. However,

a series of conservative countermeasures are taken to minimize the probability that the reliability drop below the low threshold r_{th}^{-}, which constitute *exceptions* to the general rule. Hence, no feedback is sent when the values of reliability $r[k]$ and short-term reliability $r_{sh}[k]$ are within the two thresholds, even if the value of predicted reliability $\hat{r}[k+1]$ is above the high threshold r_{th}^{+} (Exception 1 in Algorithm 4). Furthermore, when the value of predicted reliability $\hat{r}[k+1]$ is within r_{th}^{+} and r_{th}^{-} or above r_{th}^{+}, and the value of the short-term reliability $r_{sh}[k]$ is within the thresholds, but the actual value of the reliability $r[k]$ is below r_{th}^{-}, this identifies an uptrend in the reliability that needs to be consolidated and accelerated by sending a feedback that advertises low reliability $r[k]$ (Exception 2 in Algorithm 4). Finally, whenever the value of the short-term reliability $r_{sh}[k]$ drops below the threshold r_{th}^{-}, the advertised value $r_{adv}[k]$ is set equal to the short-term reliability $r_{sh}[k]$, irrespective of the value of reliability $r[k]$ and of the predicted reliability $\hat{r}[k+1]$ (Exception 3 in Algorithm 4). This is done to preemptively invert a downtrend before the reliability actually drops below r_{th}^{-}. The rule defined by Exception 3 has precedence over all the other rules.

2.3.6 Handling Voids

In geographical routing protocols devices can either work in a *greedy mode* or in a *recovery mode*. When in greedy mode, the node that currently holds the message tries to forward it towards the destination. The recovery mode is entered when a node fails to forward a message in the greedy mode, since none of its neighbors is a feasible next hop, i.e., none has a positive advance towards the destination. Usually this occurs when the node observes a *void region* between itself and the destination. Such a node is referred to as *concave* node. A packet enters the recovery routing mode when it reaches a concave node, and resumes greedy forwarding when it reaches a node that is closer to the destination than the concave node. Several schemes [36][21] have been proposed to solve this problem, that are based on *face routing* on *planar graphs*. The main drawback of these solutions is that they may

Algorithm 4 Adaptive Control Scheme

Pseudo-code executed by each actor

$r_{adv}[k]$=$\hat{r}[k+1]$

```
// General Case
```

if (isAbove($\hat{r}[k+1]$) **or** isBelow($\hat{r}[k+1]$)) **then**

  feedbackNeeded=**true**

end if

```
// Exception 1:  Slow uptrend
```

if (isAbove($\hat{r}[k+1]$) **and** isWithin($r[k]$) **and** isWithin($r_{sh}[k]$)) **then**

  feedbackNeeded=**false**

end if

```
// Exception 2:  Consolidate uptrend
```

if (isBelow($r[k]$) **and** isWithin($r_{sh}[k]$) **and** !isBelow($\hat{r}[k+1]$)) **then**

  feedbackNeeded=**true**

  $r_{adv}[k]$=$r[k]$

end if

```
// Exception 3:  Low short-term reliability
```

if (isBelow($r_{sh}[k]$)) **then**

  feedbackNeeded=**true**

  $r_{adv}[k]$=$r_{sh}[k]$

end if

if (feedbackNeeded) **then**

  sendFeedback($r_{adv}[k]$)

end if

select long detouring paths [80]. When a packet reaches a concave node, recovery routing algorithms select a left or right detour path according to predefined rules. As a result, they may select a long detour paths. This problem is known as the *blind detouring problem*. For example, the hop count of detouring paths constructed by FACE-2 [21] is in average twice that of the shortest detouring path, and with a much higher variance [80].

We combine face routing, in particular the FACE-2 algorithm, with our distributed algorithm. The objective is twofold: i) the detouring path needs to be based on paths constructed by FACE-2, thus guaranteeing delivery; ii) the path length still needs to be adjusted based on the reliability observed at the actor.

The operations of the recovery mode are as follows. Assume sensor v generates or receives a packet, and v identifies itself as a concave node in the path towards its closest actor c_v. All neighbors with *absolute positive advance* with respect to v are feasible next hops. This includes all neighbors w whose distance to their closest actor c_w is smaller than the distance between v and its closest actor c_v. If no such neighbor exists, v resorts to transmitting the packet towards its closest actor c_v through a detouring path, and thus enters the recovery mode. To accomplish this, v enters the *recovery state*, where the next hop is selected according to the rules defined in FACE-2. In the recovery state, nodes transmit at their maximum power to allow all neighboring nodes to overhear their transmissions, and packets transmitted by nodes in the recovery state contain in their header a detouring-hop number, that identifies their position in the detouring path, and which we refer to as their *virtual proximity* to the destination actor. Hence, the packet transmitted by the first concave node in the path will have virtual proximity 1, and so on increasing towards the actor. When in recovery state, a generic node v initially ignores feedback messages from the actor. At the same time, v listens to the channel for packets transmitted by neighboring nodes. Whenever a node v on the detouring path overhears a packet transmitted by a neighbor w destined to the same actor and with higher virtual proximity than its own, v flags w as a feasible next hop, since it is part of the detouring path towards c_v and has a

positive virtual advance (i.e., w has virtual proximity higher than v). Through overhearing, v thus constructs the list of neighbors that are part of the detouring path towards c_v, along with their virtual proximity. Then, the operation resumes according to the rules in Section **??**. Only, decisions are now based on virtual proximity, i.e., the relative position on the detouring path. The speed-up state selects the most advanced neighbor in the detouring path, instead of the neighbor that is physically closest to the actor, while the aggregation state selects the closest neighbor in the detouring path with positive virtual advance.

2.4 Actor-Actor Coordination: Problem Formulation

The objective of the actor-actor coordination is to select the best actor(s) to perform appropriate action on the event area. Actor-actor coordination presents several analogies with the so-called *Multi-Robot Task Allocation* (MRTA) problem encountered in robotics. In fact, a fundamental questions faced when designing cooperative multi-robot systems is "Which robot should execute which task in order to cooperatively achieve the global goal?" [44]. In this chapter we are concerned with methods for *intentional cooperation*, i.e., actors cooperate explicitly through task-related communication and negotiation. Other approaches to cooperation, such as minimalist or emergent approaches [44], where individuals coordinate their actions without explicit negotiation or allocation of tasks, are out of the scope of this chapter.

At the end of the sensor-actor coordination phase described in Section 2.3, one or multiple actors, which we denote as *collectors*, receive sensor readings from source sensors that define the *event area*. The event area corresponds to the *action area*, i.e., the area where an action is required. In particular, each collector receives data from a subset of the sources. Each *part* in the partition in Section 2.2 identifies a portion of the action/event area and is under the responsibility of the corresponding collector. However, the collector may not be able to act on the entire area that it is responsible for, since the area may not be totally within the collector's *action range*. The action range defines the circular area where

an actor is able to act. Moreover, the collector may not be the "best" actor for that task in terms of *action completion time* and/or *energy consumption*, where the former is the time to perform the action and the latter is the required energy for the action. For these reasons, actor-actor coordination is required before initiating the action.

Definition 7 *The* action completion bound *is the maximum allowed time from the instant when the event is sensed to the instant when the action is completed.*

The coordination objective of each collector actor is to find the optimal actors to timely act on the portion of the event area under its own responsibility. In particular, if multiple actors can act on a certain area we refer to the area as an *overlapping area* (region areas numbered from 1 to 8 in Fig. 4). On an overlapping area the actor-actor coordination problem consists of selecting a subset of the actors and their action powers, to optimally divide the action workload, so as to maximize the *residual energy* to extend the lifetime of the actors[4] while complying with the *action completion bound*. We refer to an area where only one actor can act as a *non-overlapping area* (unshaded regions in Fig. 4). For such an area, the coordination problem simplifies to selecting the power level for the actor that minimizes the energy consumption while abiding by the action completion bound. For this reason, we assume that the coordination problem involves only overlapping areas and that the available energy of each actor is already discounted with the energy needed to act on non-overlapping areas.

In Section 2.4.1 we introduce the action area and the actor models, which are then used in the formulation of the problem in Section 2.4.2.

2.4.1 Action Area and Actor Model

Let $\mathcal{S}^A$ represent the set of actors, with $N_A = |\mathcal{S}^A|$, and let $\mathcal{S}^C$ be the set of collectors ($\mathcal{S}^C \subseteq \mathcal{S}^A$). As mentioned before, collectors receive data from sources (sensors), and

[4]Although actors are resource-rich nodes, the order of magnitude of the energy required for actions is higher than that required for communication. Hence, it is important to save action energy in order to extend the lifetime of actors.

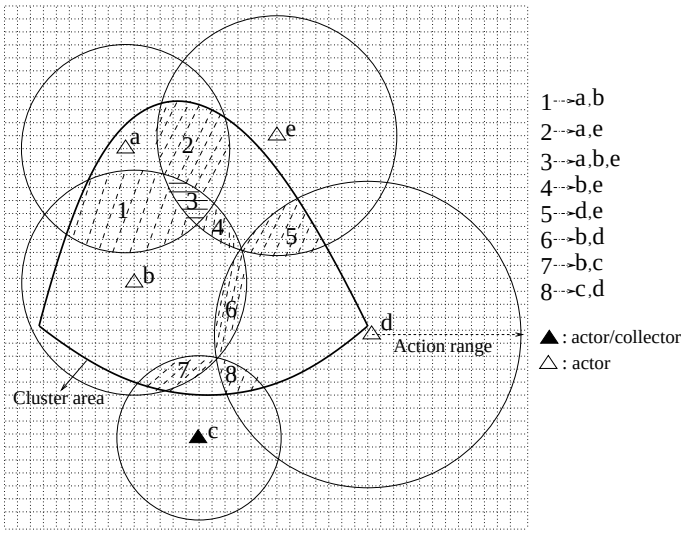


Figure 4: Overlapping and Non-overlapping areas for collector c.

from the source positions they can identify the portion of the whole event area they are responsible for. By referring to Fig. 4, we introduce the following notation:

- $\mathcal{A}^h_{c,nov}$ and $\mathcal{A}^m_{c,ov}$ are the h^{th} *non-overlapping* and the m^{th} *overlapping* areas, respectively, inside the portion of area under the responsibility of collector c. H_c represents the number of non-overlapping areas, while M_c represents the number of overlapping areas associated with collector c;

- $\mathcal{S}^{A,m}_{c,ov}$ is the set of actors that can act on the m^{th} overlapping area $\mathcal{A}^m_{c,ov}$ that is under the responsibility of collector c.

Each actor a is characterized by the following parameters:

- $R_a\ [m]$ is the action range of a;

- $P_a^{Max}\ [W]$ is the maximum power that actor a can use to perform the action. Actors can select their power among L different levels

$$P_{a,p} = \frac{P_a^{Max}}{L} \cdot p,\ p = 1, 2, \ldots, L \tag{16}$$

where $P_{a,p}$ is the p^{th} power level for actor a. As will be shown in (17), a higher power corresponds to a lower action completion time;

- η_a is the *efficiency* of actor a (see (17));

- $E_a^{Av}[J]$ is the available energy of actor a, discounted with the energy needed to act on non-overlapping areas where only actor a can act.

2.4.2 Mixed Integer Non-Linear Program

In this section we formulate the actor-actor coordination problem as a *Mixed Integer Non-Linear Program* (MINLP). The objective is to find, for each portion of the event area, the subset of actors that maximizes the average residual energy of all actors involved in the action, under the constraint of meeting the action completion bound. We define the problem according to the following assumptions: i) the energy required to perform the action is orders of magnitude higher than the energy required for communication; ii) actors are able to *selectively* act on part of their action area, i.e., if actor a is chosen to act either on an overlapping or non-overlapping area, this does not imply that it must act on the entire area in its action range.

Let us introduce the following notations:

- $P_{a,p}^{(m)}\ [W]$ is the p^{th} power level of actor a for the m^{th} overlapping area $\mathcal{A}_{c,ov}^m$, whose measure is $A_{c,ov}^m\ [m^2]$;

- $\underline{X}^{(m)}$ is a binary matrix whose element $[x_{a,p}^{(m)}]$ is equal to 1 iff actor a acts on the overlapping area $\mathcal{A}_{c,ov}^m$ using power level $P_{a,p}^{(m)}$;

- $T_{a,p}^{(m)}\ [s]$ is the action completion time for actor a acting alone and independently on the m^{th} overlapping area, when the actor uses the p^{th} power level

$$T_{a,p}^{(m)} = K \cdot \frac{A_{c,ov}^m}{\eta_a \cdot (P_{a,p}^{(m)})^{\gamma_a}}, \tag{17}$$

  where $K\ [W^{\gamma_a} \cdot s/m^2]$ is a constant, γ_a is a parameter ranging in $(0, 1]$, which defines the power-time relationship for actor a, and η_a is the actor efficiency;

- $\delta\ [s]$ is the *action completion bound* (i.e., the maximum time for the action to be completed), which depends on the event and on the application;

- $I_a^{(m)}$ is equal to 1 iff the m^{th} overlapping area is in the action range of actor a, 0 otherwise;

- h_a is a binary variable equal to 1 iff actor a is involved in an action.

We can now formulate the optimization problem as follows:

$\mathbf{P}_{Max}^{Res}$: **Residual Energy Maximization Problem**

$$Given: \quad N_A,\ L, M_c, E_a^{Av},\ T_{a,p}^{(m)}, I_a^{(m)}$$

$$Find: \qquad \underline{X}^{(m)} = [x_{a,p}^{(m)}],\ h_a \tag{18}$$

$$Maximize: \qquad E_{Avg}^{Res} = \frac{\sum_{a=1}^{N_A} h_a E_a^{Res}}{\sum_{a=1}^{N_A} h_a} \tag{19}$$

$$Subject\ to:$$

$$E_a^{Res} = E_a^{Av} - E_a^{Req} \geq 0,\ \forall a; \tag{20}$$

$$E_a^{Req} = \sum_{m=1}^{M_c} \left(\frac{\sum_{p=1}^{L} x_{a,p}^{(m)} P_{a,p}^{(m)}}{\sum_{a=1}^{N_A} \sum_{p=1}^{L} \frac{x_{a,p}^{(m)}}{T_{a,p}^{(m)}}} \right),\ \forall a; \tag{21}$$

$$\sum_{p=1}^{L} x_{a,p}^{(m)} \leq 1,\ \forall a,\ \forall m; \qquad \sum_{a=1}^{N_A} \sum_{p=1}^{L} x_{a,p}^{(m)} \geq 1,\ \forall m; \tag{22}$$

$$\frac{1}{\sum_{a=1}^{N_A} \sum_{p=1}^{L} \frac{x_{a,p}^{(m)}}{T_{a,p}^{(m)}}} \leq \delta,\ \forall m; \tag{23}$$

$$h_a \leq \sum_{p=1}^{L} \sum_{m=1}^{M_c} x_{a,p}^{(m)},\ \forall a; \qquad h_a \geq x_{a,p}^{(m)},\ \forall a,\ \forall p,\ \forall m; \tag{24}$$

$$x_{a,p}^{(m)} \leq I_a^{(m)},\ \forall a,\ \forall p,\ \forall m. \tag{25}$$

Constraint (20) guarantees a non-negative residual energy for each actor. Constraint (21) defines the energy required for actor a to complete the action on the overlapping areas where it is involved. The constraints in (22) ensure that each actor use only one among its power levels, and that at least one actor act on each overlapping area, respectively. Note that

when multiple actor act on an area, the time to complete the action is reduced. Accordingly, constraint (23) limits the overall action completion time, expressed as $\left(\sum_{a=1}^{N}\sum_{p=1}^{L}\frac{x_{a,p}^{(m)}}{T_{a,p}^{(m)}}\right)^{-1}$ for multiple actors acting on the area, to be smaller than the action completion bound, for each overlapping area. The constraints in (24) define the relation between the $x_{a,p}^{(m)}$ and h_a variables, while constraint (25) imposes that each actor act only on areas in its action range.

2.5 Actor-Actor Coordination: Localized Auction Protocol

In this section, we propose a distributed solution to the actor-actor coordination problem stated in Section 2.4. In particular, our solution is inspired by the behavior of agents in a *market economy*, where the allocation of resources occurs as a result of interactions between buyers and sellers [70][43]. Auctions have proven a powerful tool for achieving efficient resource allocations, especially in large-scale environments in which the acquisition of consistent global state information leads to excessive overhead [43]. Our approach is based on a *real-time auction protocol* that describes the behavior of actors participating in transactions as buyers/sellers. Since globally optimal allocation of resources may not be feasible or desirable in a distributed setting, we investigate a practical and efficient method for allocating tasks to heterogeneous actors, and verify its performance by comparing it to the global optimum. The objective of the auction is to select the best set of actors to perform the action on each overlapping area. Thus, overlapping areas are *items* that are traded by the actors. Actors can assume the following roles:

- *Seller.* Is the actor responsible for a portion of event area, i.e., the actor that receives event features for that area. It corresponds to a collector.

- *Auctioneer.* Is the actor in charge of conducting the auction on a particular overlapping area. It is selected for each overlapping area by the collector/seller responsible for that area.

- *Buyer.* The actors that can act on a particular overlapping area are referred to as buyers for that area.

A localized auction takes place in each overlapping area. The *bid* of each actor participating in the auction consists of a power level and of the corresponding *action completion time* (i.e., the time needed by that actor to complete the action on the whole area) defined in (16) and (17), respectively, as well as the available energy of the actor. The objective is to maximize the total *revenue* of the team, where the team is constituted by the actors participating in the auction, and the revenue depends on the *residual energy* (i.e., E_{Avg}^{Res}, in Section 2.4). *Multiple localized auctions* take place in parallel under the responsibility of different auctioneers. This is preferable to one single auction conducted by the collector for several reasons: i) it causes lower signaling overhead, since the auction messages are exchanged between the auctioneer and the buyers for that overlapping area, which are close to the auctioneer; ii) the auction process workload is shared among a higher number of actors, since the number of auctioneers is in general higher than the number of collectors; iii) it is scalable as the number of actors increases.

When seller c (the collector) receives the event features from the sensors, it decides whether an action needs to be performed on the area it is responsible for and computes all the non-overlapping and overlapping areas. The coordination problem arises for the overlapping areas where more than one actor can act, while for the non-overlapping areas the seller directly assigns the action task to the corresponding actor.

Seller c selects M_c *auctioneers*, one for each overlapping area, among the actors that can act on each of these areas. Let $s^{(m)} \in \mathcal{S}^A$ be the auctioneer selected by seller c to conduct the auction for the m^{th} overlapping area. This auctioneer is selected to be the closest actor to the center of the overlapping area. This way, since the auctioneer is close to each actor in the overlapping area, the energy spent for communication and the auction time is reduced. After selecting the auctioneer $s^{(m)}$, the seller c provides it with the area $\mathcal{A}_{c,ov}^m$ where the auction should take place, the *action completion bound* δ, and the *auction time*

bound τ_c, which is the maximum allowed time for the auction. The auctioneer determines the winners of the auction based on the bids it receives from the buyers. At the beginning of the auction, the auctioneer sends a `JOIN_AUCTION` message to all the buyers competing for the area. After a buyer a hears this announcement, it submits its available energy, E_a^{Av}, and L two-dimensional bids $\underline{b_a} = \{b_a^1, b_a^2, \ldots, b_a^L\}$, where $b_a^{(p)} = [P_{a,p}^{(m)}; T_{a,p}^{(m)}]$, $p = 1, 2, \ldots, L$, with $P_{a,p}^{(m)}$ and $T_{a,p}^{(m)}$ defined in (16) and (17), respectively. By means of these bids, the auctioneer determines the winners by calculating the optimal solution for the residual energy maximization problem $\mathbf{P}_{Max}^{Res}$ defined in Section 2.4.2. However, in this case the problem is limited to the overlapping area the auctioneer is responsible for. This way, since the bids are submitted to the auctioneer only once, signaling overhead is reduced [67]. In microeconomic theory, this auction mechanism can be classified as a *single-round sealed-bid auction* [70], where each buyer submits its bids in one shot irrespective of the bids from other buyers.

The solution proposed in this section can also be seen as a "divide and conquer" approach to the problem discussed in Section 2.4. A hard non-linear optimization problem, whose objective is to select the optimal actors to perform an action, is divided through the auction mechanism into several sub-problems associated with each overlapping area and that are solved in a localized fashion by individual actors. Finally, each "auctioneer" actor solves a smaller-scale version of the original problem. The sub-optimality of this solution comes from not solving the whole problem at once, and from solving instead several sub-problems in parallel and independently. However, as will be shown in Section 2.6, its performance is much better than that of simpler heuristics, and in general very close to the global optimum.

2.6 Performance Evaluation

In this section, we present performance results from the proposed framework. In Sections 2.6.1 and 2.6.2 we report the performance results for the sensor-actor coordination mechanism, while in Section 2.6.3 we discuss the actor-actor coordination.

2.6.1 Sensor-actor Coordination

The optimization problem presented in Section 2.2.2 was implemented in AMPL [40], and solved with CPLEX [2], which uses a branch and bound algorithm to solve mixed integer linear problems. The start-up, speed-up, and aggregation states, described in Section 2.3, were implemented in a C++ simulator, which we used to evaluate the energy consumption, and in the J-Sim simulator [7], which implements the whole protocol stack of a sensor node, from physical to application layer, including CSMA/CA MAC. All figures in this section report 95% confidence intervals. We considered several different simulation scenarios. In Scenario 1, the deployment area is circular with radius equal to $20m$. For each deployed sensor, the distance from the center of the area and the angle are uniformly distributed random variables. In Scenario 2, sensor nodes are randomly deployed in a square area of $25m$x$25m$. The event area is circular, with varying radius ranging in $[2, 12]m$ in different simulations. The epicenter of the event area is randomly selected such that the event area completely falls into the terrain. Scenario 3 is similar to Scenario 2, but the side of the square area is $100m$. Four actors are randomly deployed in each scenario. As in [51], the simulation parameters for the energy model in Section 2.2.1 are chosen to be $E_{elec} = 50nJ/bit$, $\beta = 100pJ/bit/m^{\alpha}$, $\alpha = 4$. The transmission range of sensors is set to $10m$.

Since the global network behavior depends on several application-dependent parameters, here we present results related to particular network configurations that constitute upper and lower bounds on the achievable performance. Hence, in this section we refer to start-up configuration, speed-up configuration, and aggregation configuration, as those

configurations where all nodes are in the start-up, speed-up, and aggregation state, respectively. This allows us to show the benefits of the proposed solution without depending on the choice of parameters that govern the transitions among states. Dynamic aspects are discussed in Section 2.6.2.

Figure 5 shows a comparison between the optimal solution to the event-driven partitioning problem, described in Section 2.2, and the energy consumption in the start-up, speed-up, and aggregation configuration in Section 2.3, respectively, with varying event ranges. The figure shows the overall network cost, i.e., the energy needed to transmit one bit from each source to the actors. Noticeably, the optimal solution is almost independent of the event range. This is due to two contrasting phenomena. The number of sources increases when the event range increases, leading to a potentially higher energy consumption; conversely, since more nodes are involved, aggregation can be increasingly leveraged. These two trends compensate each other leading to a flat curve. Conversely, the energy consumption in the start-up and speed-up configurations highly increases with the event range. As also shown in Fig. 5, this can be partially compensated by the aggregation state. In particular, an aggregation configuration can be reached both from a start-up configuration and from a speed-up configuration. An aggregation configuration reached from a start-up configuration leads to an almost-optimal energy consumption, while by reaching the aggregation configuration from a speed-up configuration, the energy consumption can still be decreased consistently, but not as much as in the previous case. The structure of the da-trees after the start-up/speed-up configuration somehow constrains an aggregation process based on simple logic and minimal interaction among sensors. Hence, Fig. 5 motivates the design of our distributed protocol. In fact, the distributed solution described in Section 2.3 modifies the structure of the da-trees to reach an energy configuration that is between the speed-up and the aggregation from start-up curves in Fig. 5. Depending on the required latency bound and reliability threshold, after a transient start-up configuration, a certain number of sensors will enter the speed-up/aggregation state to reach a minimum

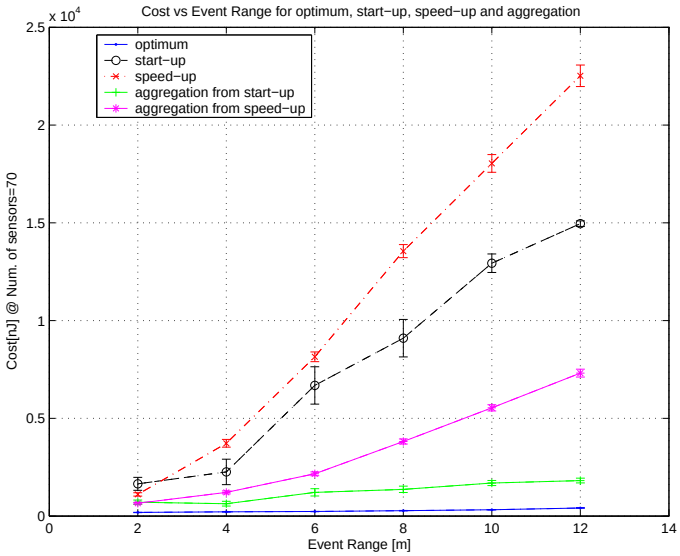


Figure 5: Scenario 1. Comparison of optimal solution, speed-up, start-up, and aggregation configuration with 70 nodes.

energy configuration, given the required reliability. When a higher reliability is required, the network will move towards a higher energy/lower delay configuration, while when the required reliability is guaranteed with some margin, the network will move towards a lower energy configuration.

In Figs. 6 and 8 we plot the average energy consumption versus the number of sensors, with different event ranges, for the start-up and aggregation configurations in Scenario 2. The energy expenditure of the aggregation configuration is two orders of magnitude lower than in the start-up configuration. As can be seen in Fig. 8, the energy expenditure increases sublinearly with the number of sensors. Figure 7 reports the overall energy consumption for the speed-up configuration. Interestingly, not only is the energy consumption of the speed-up configuration around one order of magnitude higher than in the start-up configuration; also, as already seen in Fig. 5, when the aggregation configuration is reached from a speed-up configuration, the network converges to a less energy-efficient configuration, compared to when the aggregation configuration is reached directly from the start-up configuration. This is confirmed by Fig. 9, which shows that the order of magnitude of the

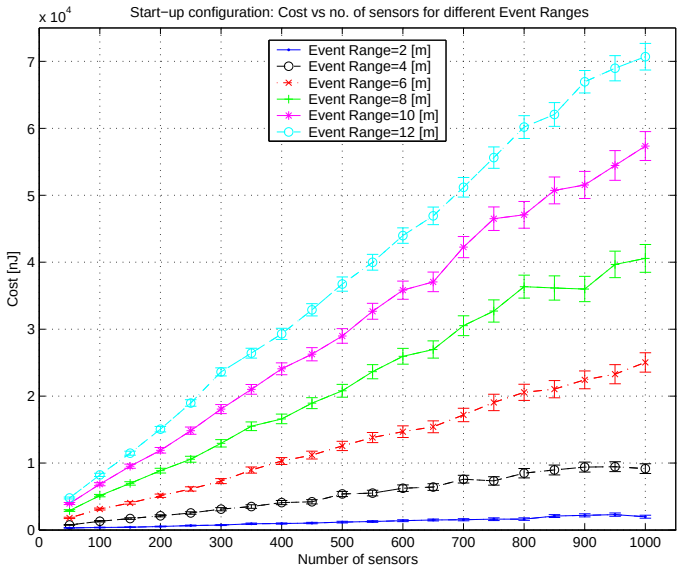


Figure 6: Scenario 2. Start-up configuration: Energy consumption vs. Number of sensors for different Event Ranges.

energy consumption is $10^4\ nJ$ for an aggregation configuration reached from a speed-up configuration. Conversely, as shown in Fig. 10, in Scenario 2 the average number of hops of each source-actor pair is reduced from around 5 hops for the start-up configuration to less than 2 hops in the speed-up configuration. The speed-up configuration leads to paths with lower delay (lower number of hops); however, since this is paid with a higher energy consumption, the speed-up mechanism should be used only when strictly necessary to provide the required reliability. A comparison of the energy efficiency of the different configurations in Scenario 2 is shown in Figs. 11 and 12.

Figure 13 shows the overall energy consumption for the start-up, speed-up, and aggregation configurations in Scenario 3, with 1000 nodes. Although the speed-up configuration can be seen to lead to a higher energy consumption, the energy consumptions of the start-up and speed-up configurations are in the same order of magnitude, i.e., the behavior of the speed-up configuration is similar to that of the start-up configuration. This happens when the transmission range of the nodes is short with respect to the distance between sensors and actors. In this case, a node in the start-up state tends to select the closest node to the

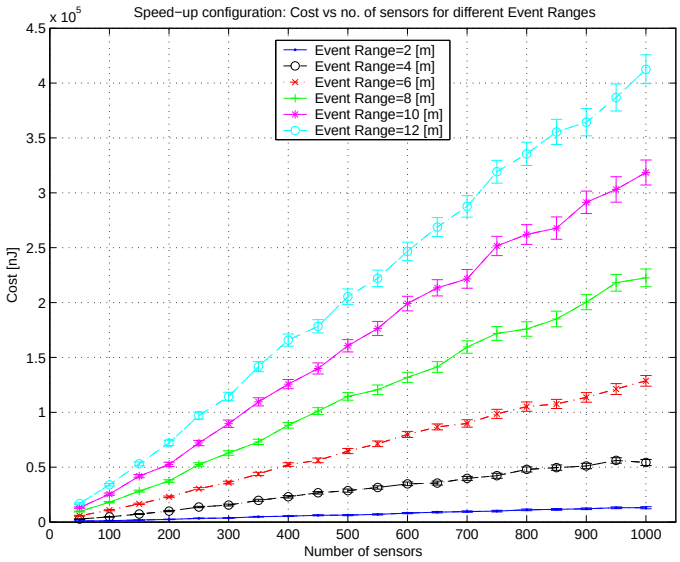


Figure 7: Scenario 2. Speed-up configuration: Energy consumption vs. Number of sensors for different Event Ranges.

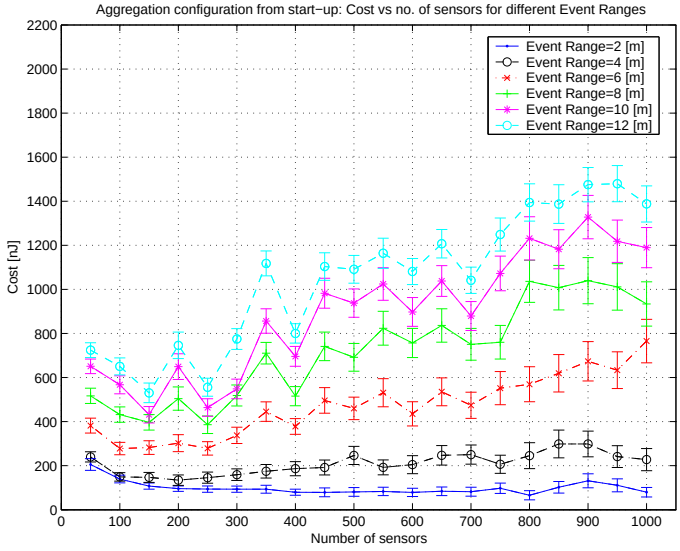


Figure 8: Scenario 2. Aggregation configuration reached from start-up configuration: Energy consumption vs. Number of sensors for different Event Ranges.

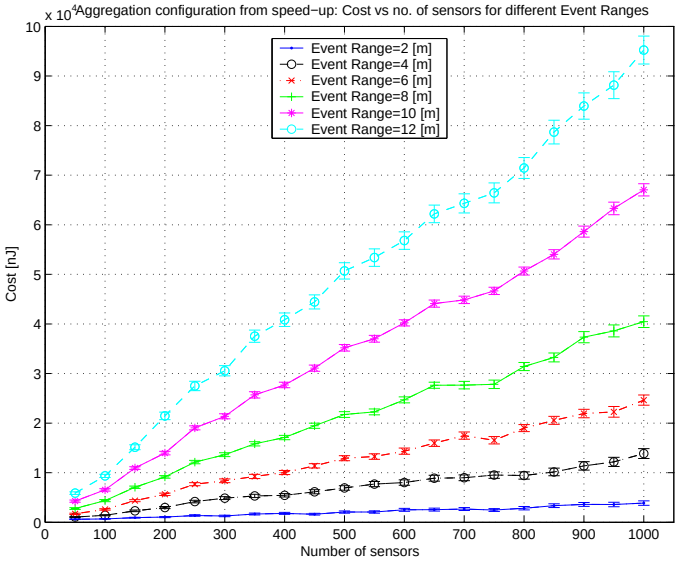


Figure 9: Scenario 2. Aggregation configuration reached from speed-up configuration: Energy consumption vs. Number of sensors for different Event Ranges.

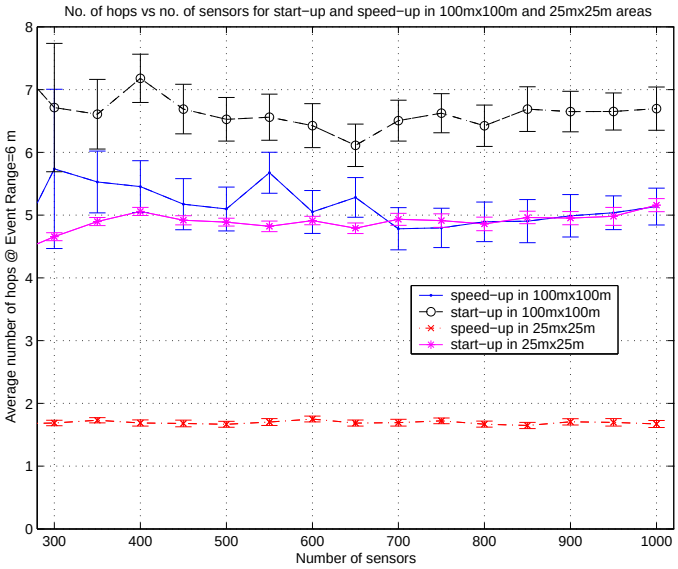


Figure 10: Scenario 2-3. Average number of hops for start-up and speed-up configurations.

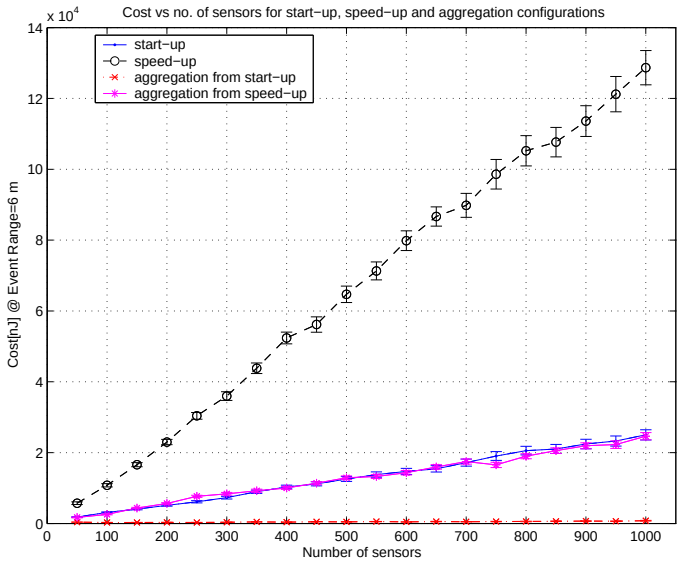


Figure 11: Scenario 2. Comparison of energy consumptions for start-up, speed-up and aggregation configurations with event range $= 6m$.

destination (as it would do in the speed-up configuration). In fact, when the distance between the sensor and the actor is much larger than the transmission range, the second term in the sum of the two-hop rule accounts for most of the energy expenditure (see Section 2.3.2). In fact, the distance between the sensor and any of its neighbors is short as compared to the distance between the neighbor and the actor. Thus, the neighbor is selected so as to minimize the second term, which, as can be easily demonstrated, results in selecting the neighbor that is closest to the destination (as also a node in the speed-up state does). By comparing Figs. 10 and 13, we can conclude that in this case the speed-up configuration still outperforms the start-up configuration in terms of number of hops, while this is achieved with a limited additional energy expenditure. This is also reflected in the distribution of packet delays. Figure 14 shows a comparison of packet delays from sensor to actor in Scenario 3, with 400 nodes, between the speed-up and the start-up configuration when the event range is set to $12m$, which corresponds to 20 sources in average. Sources generate 2 packets per second for 200 seconds of simulation. The packet size is 56 bytes, including 10 byte payload, 12 bytes of network-layer header and 34 bytes of MAC-layer

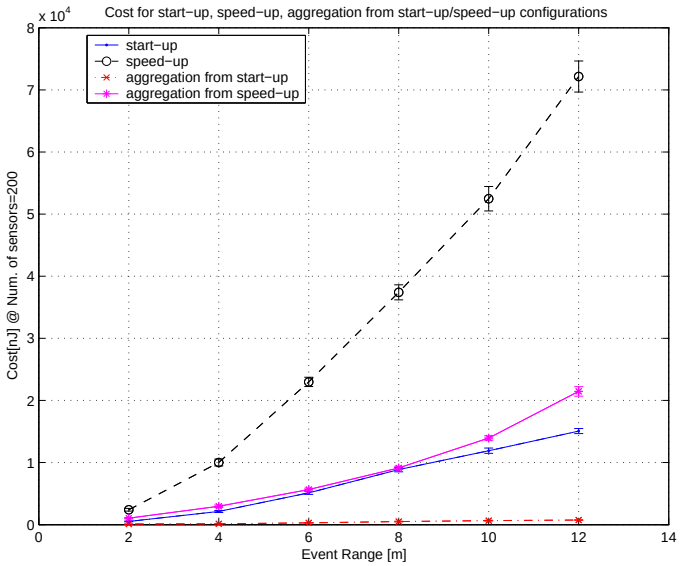


Figure 12: Scenario 2. Comparison of energy consumptions for start-up, speed-up and aggregation configurations with 200 sensors.

header. Figures 14(c) and 14(d) show the distribution of delays. In the speed-up configuration (Fig. 14(d)), the delay is below $0.5s$ for almost 100% of the packets, while in the start-up configuration (Fig. 14(c)) the variability of delays is much higher and their value can be as high as $2.5s$. Figures 15(a) and 15(b) refer to the same scenario, where each source generates 5 packets per second, for the start-up and speed-up configurations, respectively. Noticeably, while in the start-up configuration the network is congested (Fig. 15(a)), leading to extremely high values for the delays, this does not happen in the speed-up configuration, where the delays are shown to be almost always within $1s$ (Fig. 15(b)). Note that, since the latency bound is application dependent, in these simulations we do not drop packets at intermediate nodes.

2.6.2 Convergence of DEPR

In this section we discuss the convergence of the DEPR mechanism for sensor-actor coordination, introduced in Section 2.3.5. The mechanism was implemented in J-Sim [7]. Each sensor in the event area is a CBR source that generates 10 packets per second. The packet

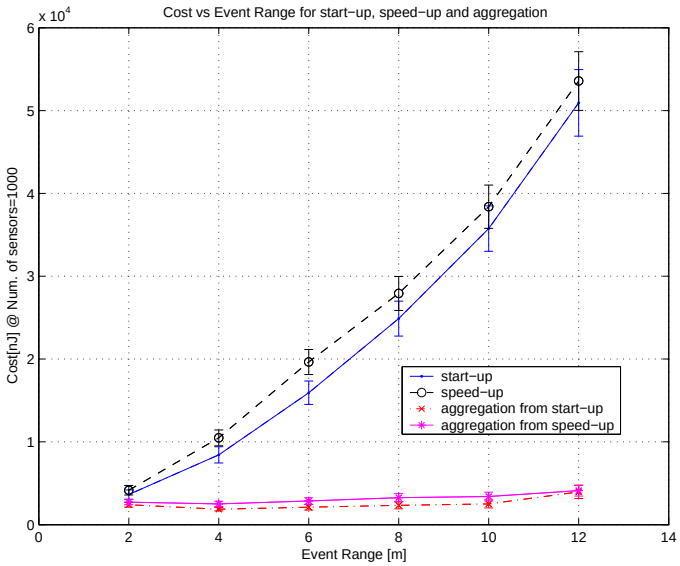


Figure 13: Scenario 3. Comparison of energy consumptions for start-up, speed-up, and aggregation configurations with 1000 sensors.

size is 56 bytes. At the network layer, sensors behave according to the DEPR protocol described in Section 2.3. The MAC layer is based on CSMA/CA, while the physical layer in J-SIM is enhanced with the power control mechanism described in Section 2.3. In this simulation, 100 sensors are randomly deployed in a $100m$x$100m$ terrain. The simulation parameters are reported in Table 1. The maximum transmission range is set to $40m$, the capacity of the channel is set to $400Kbit/s$, while the interface queue length is set to 20 packets. The event radius is equal to $15m$ and is centered at the center of the simulation area. Each sensor in the event area generates 10 packets per second. The size of the packets is 56 bytes. We evaluate the mechanism from the perspective of one actor that is placed in the middle of the lower side of the deployment terrain.

As far as the feedback mechanism is concerned, in the experiments performed we implemented a linear predictor, which calculates the predicted reliability as $\hat{r}[k+1] = \sum_{i=0}^{R-1} a_i \cdot r[k-i]$, with $R = 2$, $a_0 = 2$ and $a_1 = -1$. A thorough analysis of the impact of different predictors on the convergence of the proposed mechanism is out of the scope of this thesis. The feedback period Δf is set to $1s$, the decision interval Δd is set to $5s$ while

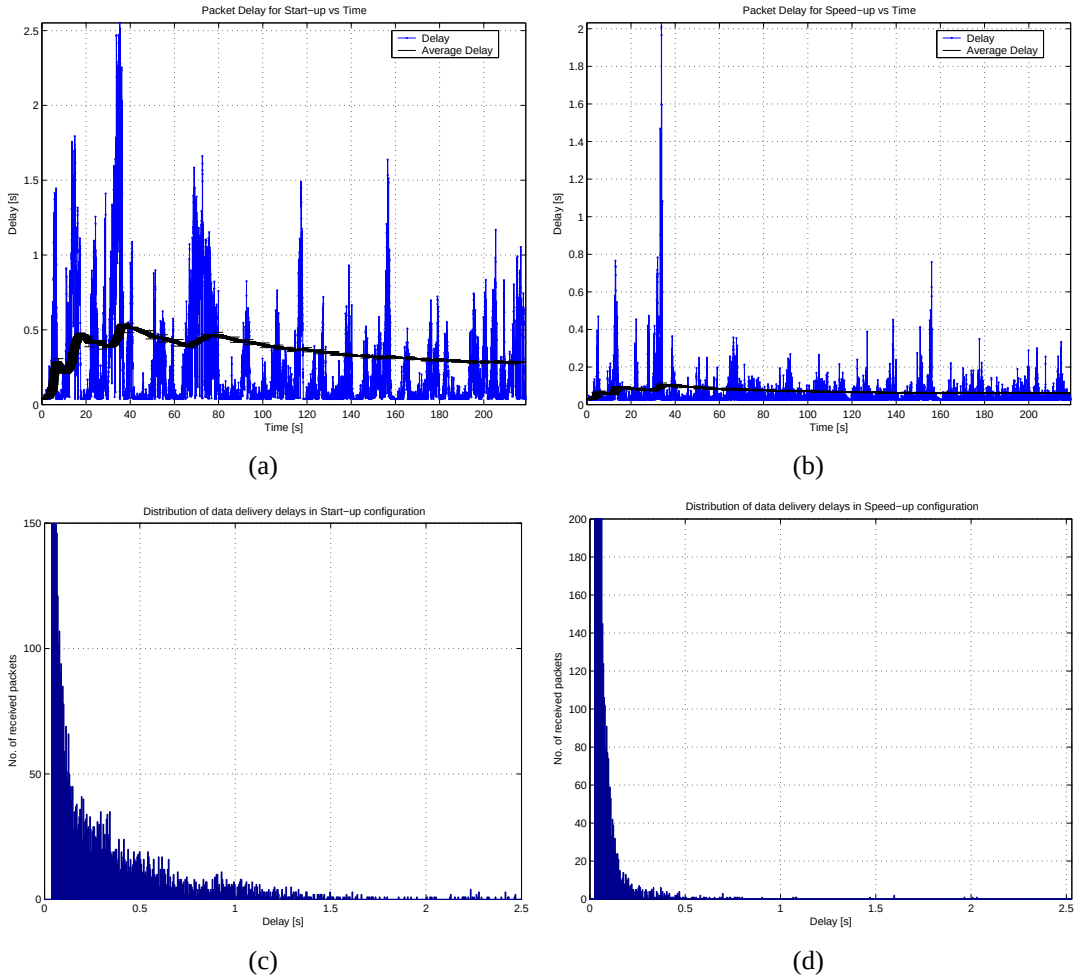


Figure 14: Scenario 3: Delays vs time (a-b) Distribution of delays (c-d) for start-up and speed-up configurations, 400 sensors, event range = $12m$, sources generating 2 packets/s.

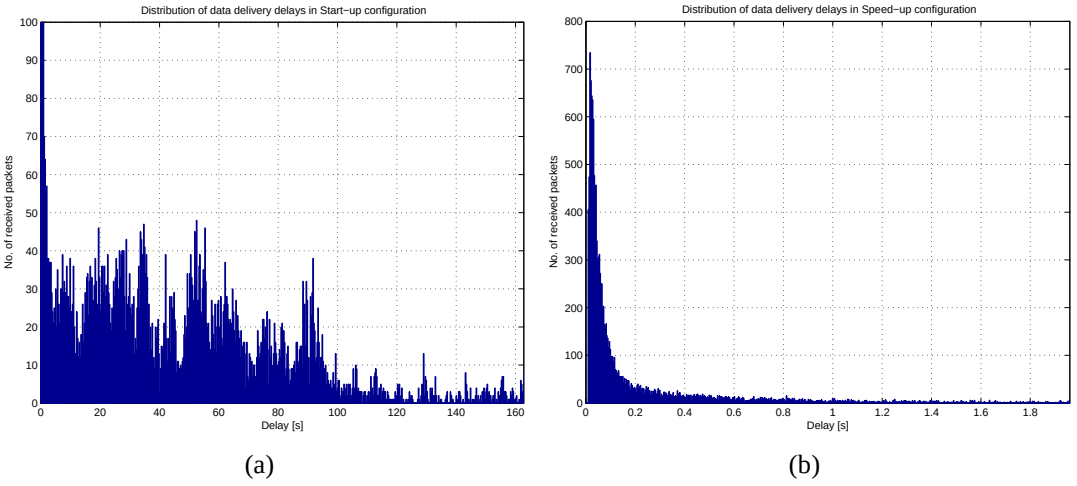


Figure 15: Scenario 3: Distribution of delays (a-b) for start-up and speed-up configurations, 400 sensors, event range = $12m$, sources generating 5 packets/s.

Table 1: Simulation Parameters - Feedback Mechanism

Parameter	Value
area	$100\,\mathrm{m}\,x100\,\mathrm{m}$
sensors	100
maximum tx range	$40\,\mathrm{m}$
bandwidth	$250\,\mathrm{kbit/s}$
packet size	$56\,\mathrm{bytes}$
queue	$20\,\mathrm{packets}$
reporting rate	$10\,\mathrm{packet/s}$
event radius	$15\,\mathrm{m}$

the short decision interval Δd_{sh} is set to $1s$. The reliability threshold is set to $r_{th} = 0.80$, while the high and low reliability thresholds are set to $r_{th}^{+} = 0.90$ and $r_{th}^{-} = 0.78$, respectively. An asymmetric tolerance zone around the reliability threshold r_{th} has in fact been shown by simulation to guarantee a low probability that the reliability drop below the low threshold r_{th}^{-}. The delay bound is set to $200ms$, which is a reasonable value for several monitoring applications.

A challenging problem is the reliable multicast support for the actor feedback messages. One simple approach is to repeatedly use the unicast sequence of RTS/CTS/DATA/ACK for reliable transmission to all recipients. However, it serializes the transmissions and increases the energy consumption of the involved nodes [37]. Furthermore, later transmitted copies may experience longer delays. The other extreme approach is to simply use the broadcast nature of the shared medium by transmitting a packet without RTS/CTS and ACK. If all the designated recipients can hear the packet successfully, all the copies received by the recipients can progress in parallel along multiple paths. However, without RTS/CTS and ACK, the probability of delivery success is low. In [37], the authors try to keep a balance between these two extremes, by selecting one of the recipients as the primary recipient. Since the routing is performed based on the geographic information, it is expected that there is high correlation among the locations of the frame recipients and thus a single CTS frame from the primary recipient provides a solution to the hidden node problem for most

recipients with a high probability. In the experiments shown, we assume ideal feedback, i.e., sensor nodes receive feedback from the actors reliably and without delay. This is to decouple the analysis of the convergence of DEPR from the particular multicast mechanism adopted. This also models accurately the situation where sensor nodes have a different radio on a different frequency to receive beacons from the actors. We have also performed experiments to assess the effect of unreliable broadcasts and of transmission delays on the mechanism, where feedback messages are transmitted by the actors and relayed by intermediate sensors until they are received by all devices in the da-trees. As expected, in this case delays prevent the reliability from asymptotically stabilizing to the desired reliability threshold value, as in the ideal case. However, although small fluctuations occur, the average reliability in general lies within the high and low reliability thresholds, and its minimum does not fall below r_{th}^{-}.

As previously discussed, our design of DEPR includes several parameters, which flexibly allow to adapt its behavior. In general, parameter tuning is either based on analytical models of the system or done by simulation. Since analytical models that accurately model the delay of large-scale wireless sensor networks under different conditions are still largely missing or are based on very restrictive assumptions, we performed it by simulation. Note that for the sake of simplicity the parameters can also be grouped, e.g., $P_{st-ag} = P_{sp-ag} = P_{ag}$ or $r_{th}^{+} = r_{th}^{-}$, so as to reduce the tuning time. The probabilities that govern the transitions among different states are set as follows. The probability P_{st-sp} of moving from the startup state to the speedup state is set to 0.5 when the advertised reliability $r < r_{th}^{-} - (0.1 \cdot r_{th}^{-})$ (very low reliability). Otherwise, if the reliability is low but close to the threshold, we try to smoothly increase the reliability and set $P_{st-sp} = 0.1$. The probability P_{st-ag} and P_{sp-ag} of moving to the aggregation state from the start-up and speed-up states, respectively, are equally set to 0.05 if the advertised reliability is equal to 1, 0.02 otherwise. In any case, the probability of switching into the aggregation state needs to be

low (less than 0.1), as higher values almost invariably cause instabilities, provoking sudden drops of the observed reliability in the transients. Finally, the transition probabilities P_{ag-sp} from the aggregation to the speed-up state, is set to 0.2 if the current, predicted, and short-term reliability are all below the threshold r_{th}^{-}. Similarly, P_{ag-sp} is set to 0.1 if only the short-term and predicted reliability are below the threshold while the current reliability is still above, while it is set to 0.05 if only the short-term reliability is below the threshold, while the others are still above. With such tuning of the parameters our objective is to minimize the probability that the reliability drop below the threshold, and still converge as quickly as possible to a lower energy configuration.

Figure 16 shows the event reliability as observed by the actor in the simulation scenario. Immediately after the initial phase, when nodes establish data paths to the actor in the start-up state, the reliability drops below the threshold. Hence, the actor advertises low reliability and a high number of sensors start moving to the speed-up state. This increases the reliability above the threshold, which in turn causes a small portion of the sensor nodes to move to the aggregation state. After a few oscillations, the reliability stabilizes at the desired value, i.e., within the high and low reliability threshold. Figure 17 shows the evolution of the number of sensors in each of the three active states. Figure 18 shows the distribution of the delays during the simulation time, while Fig. 19 shows the evolution of the delays during the simulation time. As expected, higher delays are encountered during periods of lower reliability and viceversa.

2.6.3 Actor-actor Coordination

In this section, we discuss some performance results of the actor-actor coordination problem defined in Section 2.4. The model of the MINLP problem was implemented in AMPL and solved with the MINLP solver available through the NEOS Optimization Server [31].

In Figs. 20 and 21, we compare the average residual energy with three different solution approaches, namely, the *optimal* (Section 2.4.2), *1-actor*, and *localized auction* (Section

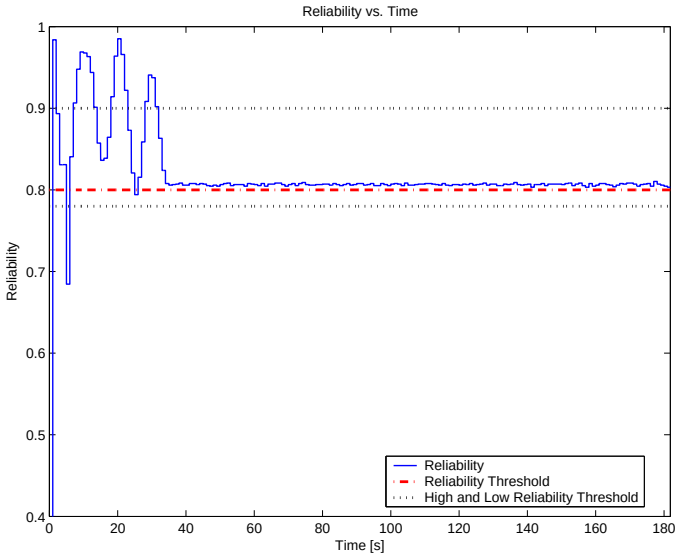


Figure 16: Reliability of the event observed at the collector/actor.

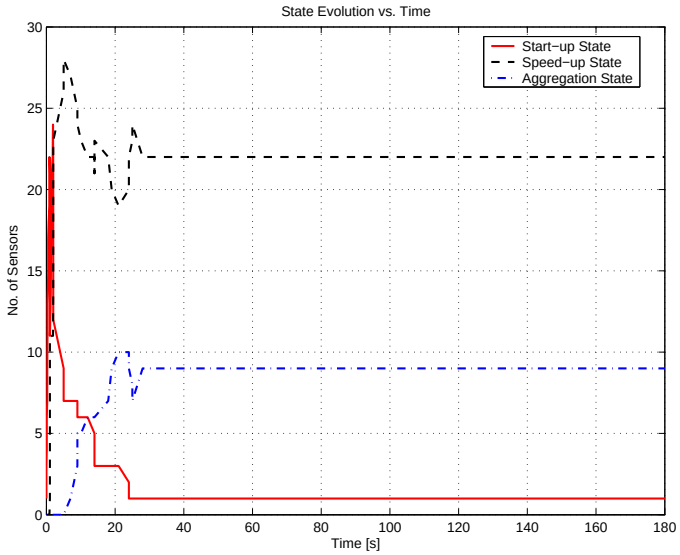


Figure 17: Number of sensors in each state with increasing simulation time.

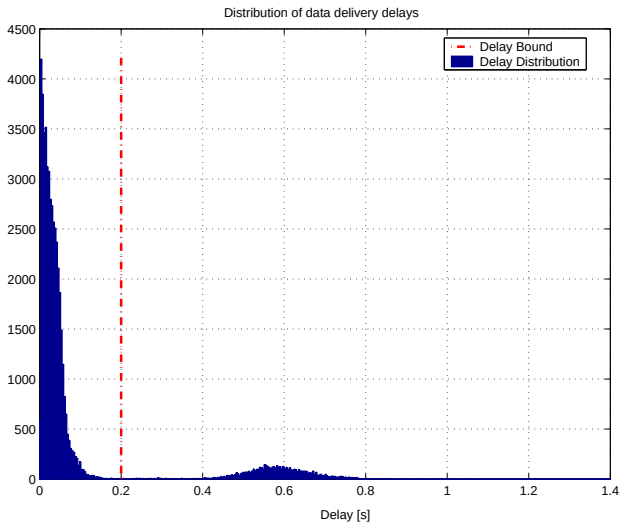


Figure 18: Distribution of delays.

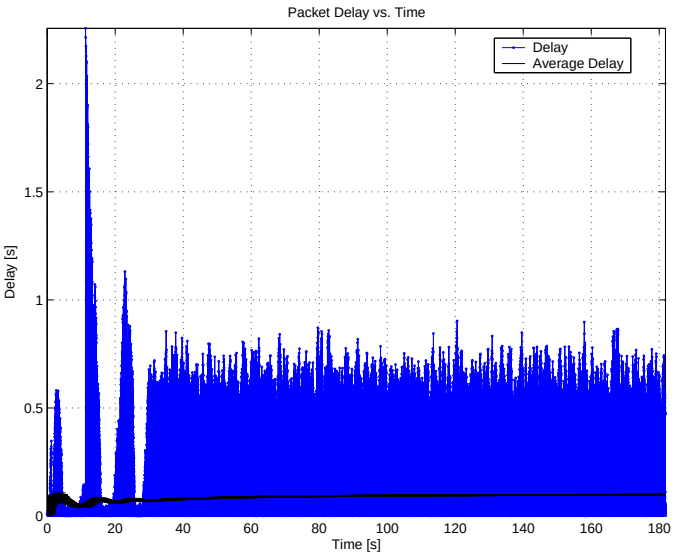


Figure 19: Delays with simulation time.

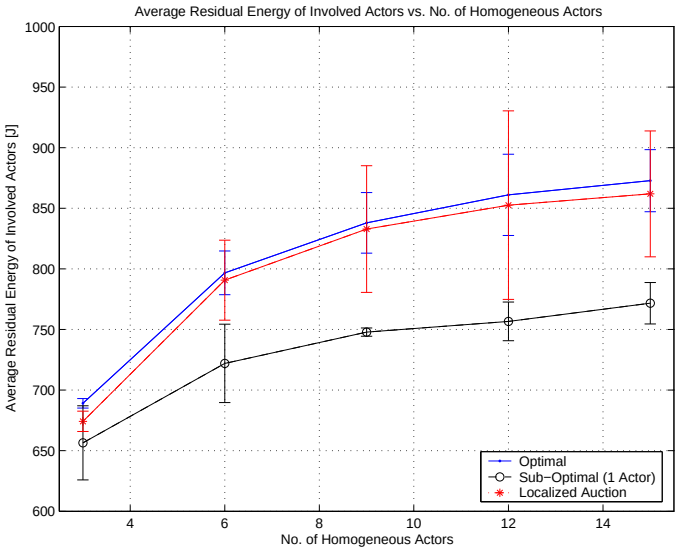


Figure 20: Average residual energy of involved actors in the homogeneous case.

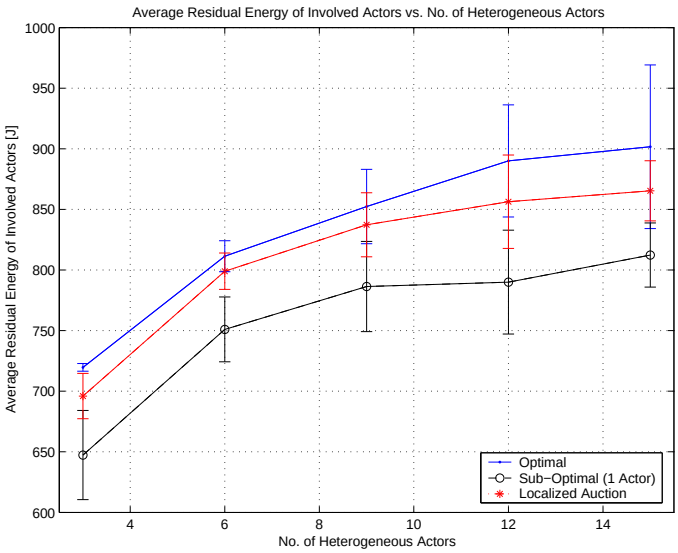


Figure 21: Average residual energy of involved actors in the heterogeneous case.

Table 2: Actor-actor Coordination Simulation Parameters

Parameter	Scenario 1	Scenario 2
$A_{c,ov}^{C,1}$; $A_{c,ov}^{C,2}$;$A_{c,ov}^{C,3}$ $[m^2]$	50;100;150	50;100;150
γ_a	0.8	0.6 - 0.9
L	5	5
K/η_a $[W^{\gamma_a} \cdot s/m^2]$	1	1
δ_c $[s]$	10	10
$\mathcal{E}_a^{in}$ $[J]$	[800,1000]	[800,1000]
P_a^{Max} $[W]$	100	100

2.5). In the optimal solution, the best set of actors is chosen so that the average residual energy of the involved actors is maximized, while guaranteeing that the action is completed before the action completion time. In the 1-actor heuristic, the action is performed by one actor only for each overlapping area, i.e., the actor with the highest residual energy after the completion of the action. In the localized auction each overlapping area is taken care of by an auctioneer that divides it among the actors based on their bids (see Section 2.5).

In the experiments performed, we concentrate on two scenarios with three overlapping areas, one with homogeneous actors with $\gamma_a = 0.8$ (Fig. 20), and one with heterogeneous actors half of which with $\gamma_a = 0.6$ (low-efficiency actors) and the other half with $\gamma_a = 0.9$ (high-efficiency actors) (Fig. 21). For the remaining parameters defined in Section 2.4 we assume the following values: $A_{c,ov}^1 = 50m^2$, $A_{c,ov}^2 = 100m^2$, $A_{c,ov}^3 = 150m^2$, $P_a^{Max} = 100W$, $L = 5$, $K/\eta_a = 1W^{\gamma_a} \cdot s/m^2$, and $\delta = 10s$. The value of the initial available energy E_a^{Av} of an actor is a random variable uniformly distributed between $800J$ and $1000J$.

As shown in both Figs. 20 and 21, the localized auction mechanism leads to near-optimal residual energy, as each auctioneer calculates the optimal solution separately for its overlapping area. However, this greatly simplifies the problem and can be achieved with local communications among actors. Moreover, in the heterogeneous scenario, the proposed localized solution effectively exploits the high-efficiency actors, thus reducing the dissipated energy to complete the action.

CHAPTER III

HANDLING MOBILITY IN WIRELESS SENSOR AND ACTOR NETWORKS

3.1 Preliminaries

As an abstraction of several application setups encountered in applications described in Section 1.1, in this chapter we refer to a scenario where sensors monitor a given terrain, and send samples of the event to the actors deployed on the terrain whenever an event occurs. Actors distributively reconstruct the event based on partial information available at different actors, estimate the event characteristics and identify an *action area*. Based on this, actors collaboratively decide on which actors should move to the action area and at which speed. The coordinated mobility of actors is thus triggered by the occurrence of events. Actors keep receiving event data until the event is active, and multiple consecutive events trigger subsequent reassignment of tasks among the actors.

In Chapter 2, we proposed a framework for communication and coordination problems with static WSANs. The concepts of *sensor-actor coordination* and *actor-actor coordination* were introduced, and centralized optimal solutions and distributed heuristics were proposed. However, many challenging applications require support for mobile actors, which is not provided in Chapter 2. Hence, in this chapter we extend our previous work in several directions.

First, we introduce a hybrid location management scheme to handle the mobility of actors with minimal energy expenditure for the sensors. The proposed solution is tailored for WSAN applications and overcomes the drawbacks of previously proposed localization services [65][32]. Actors broadcast updates limiting their scope based on Voronoi diagrams,

while sensors predict the movements of actors based on Kalman filtering of previously received updates. Our proposed scheme combines joint use of Kalman filtering with Voronoi scoping on sensors and actors to lead to a new location management technique, which is shown to consistently reduce the energy consumption on sensors by avoiding over 75% of location updates with respect to existing location update algorithms.

The second contribution of this chapter is the development of an integrated routing-physical layer scheme for sensor-actor communication based on geographical routing, which is suited for mobile WSANs. We derive a simple yet optimal forwarding rule based on geographic position in presence of Rayleigh fading channels. With respect to previously proposed geographic forwarding rules [98][93], our rule is optimal from the energy consumption standpoint. Furthermore, we show how to control the delay of the data-delivery process based on power control, i.e., to trade optimal energy consumption for decreased delay in case of low or moderate traffic. In case of high traffic, we introduce a new network congestion control mechanism at the network layer that forces multiple actors to share the traffic generated in the event area. This is shown to reduce delay, packet drops, and energy consumption even when traffic is sent to actors that are suboptimal from a network layer standpoint.

As a last contribution in our proposed system architecture, a new model for actor-actor coordination is introduced that enables coordinating motion and action of the participating actors based on the characteristics of multiple, concurrent events. In particular, it selects the best actor(s) to form the actor team to perform the required actions, based on the characteristics of the event, and drives the motion of the team towards the relevant area.

The chapter is organized as follows. In Section 3.2, we describe the proposed location management scheme, while in Section 3.3, we describe the sensor-actor communication solution. In Section 3.4, we introduce the actor-actor coordination model. In Section 3.5, we present performance evaluation results.

3.2 Location Management

The network is composed of N_S sensors and N_A actors, with $N_S >> N_A$. While sensors are densely deployed, i.e., on the order of hundreds or thousands, actors are loosely deployed. Each sensor is equipped with a low data rate radio interface, which we refer to as *sensor interface*. Actors are equipped with two radio transmitters, i.e., a low data rate transmitter to communicate with the sensors, and a high rate wireless interface for actor-actor communication. (e.g., IEEE 802.11g). We refer to these two interfaces as the actor's *sensor interface* and *actor interface*, respectively. Actors communicate among them based on a (possibly multi-hop) *ad hoc network* realized by wirelessly interconnecting their actor interfaces. From a communication perspective, actors are not energy constrained to the same extent to which sensors are. In fact, actors are sophisticated devices whose major sources of energy consumption are the actuation capabilities that allow them to move and interact with the physical world. Hence, to a first approximation, the performance of the ad hoc network among the actors is not crucial as there exist standards such as IEEE 802.11g that can provide a communication infrastructure that satisfies the performance requirements of actor-actor communication at reasonable cost as compared to the hardware needed to enable motion and actuation (motors, servos, CPUs, cameras, etc). From the perspective of sensors, actors are *equivalent recipients of information*. Hence, each actor can indistinctively be a recipient for the information generated by a sensor. Hence, each sensor will try to route information to its closest actor, unless an alternative actor is preferable in case of congestion, as described later.

In line with recent work on routing algorithms for sensor networks [75][74][93][98], we study the *sensor-actor coordination* based on a geographical routing paradigm. Geographical routing algorithms are attractive especially for their *scalability*, as it is possible to scale the network size without increasing the signaling overhead, because routing decisions are inherently *localized* [74]. The scalability of geographical routing protocols is apparent in static sensor networks with a single sink. In networks with mobile nodes and

multiple recipients, however, it depends on the availability of efficient location management schemes that are able to provide relevant nodes with the position of mobile nodes at any time. Previous proposals have dealt with the development of scalable location services for tracking mobile nodes in distributed systems based on geographical routing. In [65], GLS was proposed, which is a hierarchical location service where each mobile node maintains its current location in a number of location servers distributed throughout the network. The location servers for each node are determined based on a hashing function in the node identifier space. In [32], the performance of GLS is compared to two other location services based on similar premises. In general, the objective of these mechanisms, which can be classified as rendezvous-based protocols [32], is to potentially allow each single device in the network to retrieve the location of any other node, based on queries and replies. Clearly, query-based mechanisms can introduce delays that may not be acceptable in delay-critical systems such as WSANs. Moreover, the extensive message exchange and complex server structures, often hierarchical, associated with these protocols, can be avoided given the characteristics of WSANs.

In general, location management may follow two strategies: *location updating* and *location prediction*. Location updating is a passive strategy in which each actor periodically broadcasts its position to the neighboring sensors. The tracking accuracy depends then on the frequency of these actor-initiated updates. Location prediction is a dynamic strategy in which sensors proactively estimate the location of their neighboring actors. In this case, the tracking efficiency depends on the accuracy of the mobility model and on the efficiency of the prediction algorithm. Our proposed solution is based on a hybrid scheme. The underlying principle is to leverage the characteristics of WSANs to minimize location updates in the spatial and temporal domains, since every location update causes energy consumption at the receiving sensors, and may lead to the *broadcast storm* problem when update messages need to be relayed throughout the network. For this reason, we propose a proactive location management approach based on update messages sent by mobile actors to sensors.

As discussed, in WSANs each actor is an equivalent recipient of information. Therefore, sensor-actor communications are localized, i.e., each sensor sends information to its closest actor. Hence, in the spatial domain, broadcasts can be limited based on Voronoi diagrams [17]. At the same time, actor movement is to some extent predictable, as it is driven by the actor-actor coordination procedures. Hence, in the temporal domain, location updates can be limited to *actor positions that cannot be predicted* at the sensor side. Location updates are triggered at the actors when the actual position of the actor is "far" from what can be predicted at the sensors based on past measurements. Therefore, actors that move following predictable trajectories, which is likely to be a common case in WSANs, will need to update their position much less frequently than actors that follow temporally uncorrelated trajectories.

It may be objected that an easier way for sensors to know the future position of the actor is to let the actor itself indicate its intended destination. Sensors could then rely on this information to determine the actor trajectory. However, this is not a conclusive solution. As extensively studied in the vast literature in the field [64], motion planning in robotics is a complex problem. First, there could be multiple paths to the destination. More importantly, even the actor cannot exactly know its precise future movement patterns, as they depend on several factors such as environment inputs, reading from sensors, and coordination with other actors. For this reason, the trajectory of actors may diverge from its stated path to dynamically adjust to an evolving environment. Hence, the sensor network needs to dynamically adapt to such unanticipated trajectory changes.

In Section 3.2.1, we describe the location update strategy, while in Section 3.2.2 we describe the location prediction strategy.

3.2.1 Limiting Broadcasts in Space

A fundamental data structure from computational geometry, i.e., the Voronoi diagram, can be used in sensor actor networks to limit the scope of actor-initiated location updates. The

Voronoi diagram has been studied and applied to many domains. The work in [17] provides an extensive survey of theory and applications of Voronoi Diagrams. The Voronoi diagram of a set of discrete sites (points) on a plane partitions the plane into a set of convex polygons such that all points inside a polygon are closest to only one site. This construction effectively produces polygons with edges that are equidistant from neighboring sites.

For their properties and ease of computation, Voronoi diagrams have also been previously applied to the area of sensor networks. For example, in [72], they are used along with Delaunay triangulation to study sensor network coverage. Conversely, we rely on Voronoi diagrams to limit the spatial extension of actor broadcasts.

Formally, let $\mathcal{A} = \{a_1, a_2, .., a_{N_A}\}$ represent a set of deployed actors. For two distinct actors $a_i, a_j \in \mathcal{A}$, the *dominance* of a_i over a_j is defined as the subset of the plane being at least as close to a_i as to a_j, i.e.,

$$dom(a_i, a_j) = \{x \in R^2 | \delta(x, a_i) \leq \delta(x, a_j j)\}, \tag{26}$$

where $\delta()$ denotes the Euclidean distance. The set $dom(a_i, a_j)$ denotes a closed half plane bounded by the perpendicular bisector of a_i and a_j, which is denoted as *separator* of a_i and a_j. The Voronoi *cell* of an actor a_i is the portion of the plane that lies in all the dominances of a_i over all the other actors in $\mathcal{A}$, i.e.,

$$cell(a_i) = \bigcap_{a_k \in \mathcal{A} \setminus a_i} dom(a_i, a_k) \tag{27}$$

The Voronoi cell of an actor a_i contains all points of the plane that are closer to a_i than to any other actor in the network. We introduce the following definition:

Definition 8 *A sensor s_i is said to be* dominated *by actor a_j if its location lies in $cell(a_j)$, i.e., in the Voronoi cell of a_j.*

Every actor is responsible for location updates to sensors on its Voronoi cell, and regulates its power so as to avoid interfering beyond the farthest point in its Voronoi cell.

Hence, each sensor will expect to receive location updates from the actor it is dominated from. With respect to naive strategies such as flooding, the energy consumption for location updates is drastically reduced. It can be easily shown that the worst-case energy consumption of a flooding scheme increases as a function order of $O(N_S^2 \cdot N_A)$, and most of the energy burden is on sensors. Conversely, by limiting the scope of the location updates to the Voronoi cells created by the actors, if the actor is able to reach all sensors in its cell in one hop, which may be true in many practical cases, the energy consumption increases as a function order of $O(N_S)$.

3.2.2 Limiting Broadcasts in Time

3.2.2.1 Movement Model

The dynamic movement model for the i^{th} actor in two-dimensional Cartesian coordinates can be described by a continuous time linear dynamical system

$$\dot{\mathbf{x_i}}(t) = \mathbf{F}\mathbf{x_i}(t) + \mathbf{G}\mathbf{u_i}(t) + \mathbf{B}\mathbf{w_i}(t), \tag{28}$$

where

$$\mathbf{F} = \begin{bmatrix} 0 & \mathbf{I} \\ 0 & 0 \end{bmatrix}, \mathbf{G} = \begin{bmatrix} 0 \\ \mathbf{I} \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 \\ \mathbf{I} \end{bmatrix}, \mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{29}$$

The vector $\mathbf{x_i}(t) = [x_i(t), y_i(t), \dot{x}_i(t), \dot{y}_i(t)]^T$ represents the state of actor i at time t, i.e., its position and velocity on the two dimensional plane. Similarly, vector $\mathbf{u_i}(t) = [u_i^x(t), u_i^y(t)]^T$ represents the control input (acceleration) for actor i at time t, while $\mathbf{w_i}(t) = [w_i^x(t), w_i^y(t)]^T$ represents the process noise, i.e., unknown random acceleration on the actor trajectory.

The equivalent discrete-time dynamic equation can be derived as in [69] by means of the state space method. Hence,

$$\mathbf{x_i^k} = \mathbf{F}\mathbf{x_i^{k-1}} + \mathbf{G}\mathbf{u_i^{k-1}} + \mathbf{B}\mathbf{w_i^{k-1}} \tag{30}$$

represents the state transition equation for the system describing the motion of actor i between steps $k-1$ and k. Here, $\mathbf{x_i^k} = [x_i^k, y_i^k, \dot{x}_i^k, \dot{y}_i^k]^T$ represents position and velocity of actor i at step k, $\mathbf{u_i^k} = [u_i^{k,x}, u_i^{k,y}]^T$ represents the control input for $t \in [kT, (k+1)T)$, where T is the sampling interval, and $\mathbf{w_i^k} = [w_i^{k,x}, w_i^{k,y}]^T$ represents the discrete random acceleration. The variable $\mathbf{w_i^k}$ represents two dimensional samples of discrete time white Gaussian noise. Hence, $\mathbf{w_i^k} \sim \mathcal{N}(0, \mathbf{Q})$, with $\mathbf{Q} \geq 0$, where $\mathbf{Q}$ is the covariance matrix of the process. The random acceleration is also assumed to be independent on the two axes. $\mathbf{F}$ is called the state transition matrix, and is defined as in (29).

The state $\mathbf{x_i^k}$ at step k can be observed by the actor. The observation is related to the state by the *measurement equation*

$$\mathbf{z_i^k} = \mathbf{H}\mathbf{x_i^k} + \mathbf{B}\mathbf{v_i^k} \tag{31}$$

where $\mathbf{z_i^k} = [z_i^{k,x}, z_i^{k,y}]$ represents the *observed position* of the actor at step k, and where

$$\mathbf{H} = \begin{bmatrix} \mathbf{I} & 0 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 0 \\ \mathbf{I} \end{bmatrix}. \tag{32}$$

The variable $\mathbf{v_i^k} = [v_i^{k,x}, v_i^{k,y}]^T$ describes the *measurement noise*, expressed as two-dimensional samples of discrete time white Gaussian noise. Hence, $\mathbf{v_i^k} \sim \mathcal{N}(0, \mathbf{R})$, with $\mathbf{R} \geq 0$, where $\mathbf{R}$ is the covariance matrix of the process. The observed position of the actor $\mathbf{z_i^k}$ is thus the actual position of the actor affected by a measurement noise, which we represent as a Gaussian variable. It is worth pointing out that we keep the model general, i.e., we do not model a particular localization technique for the actor. In fact, depending on the application setup, there are several techniques to allow the actors to determine their position. For example, in outdoor settings the actors may be equipped with GPS (Global Positioning System) receivers. Several other techniques have been employed in the robotics community to allow robots to determine their position based on measurements taken by onboard sensors, such as particle filtering [92]. Furthermore, measurements taken by the sensor networks can be used by the actors to calculate their position. Here, irrespective of how

the position is actually calculated, we are interested in how the measurement errors may affect the communication process, and on how to decrease the energy consumption of the sensor network by accurately controlling the scope and amount of localization information transmitted by the actors to the sensors.

3.2.2.2 Actor-side Position Estimation

The model presented in the previous section describes a discrete-time linear process with Gaussian noise. The Kalman filter [22] provides a computationally efficient set of recursive equations that allow estimating the state of such process. Under these assumptions, the Kalman filter can be proven to be the optimal filter in the minimum square sense, i.e., it minimizes the mean of the squared estimate error. Interestingly, as will be shown in what follows, Kalman filtering techniques can be used to *estimate the position* at the actor based on measurements, and to *predict* the position of the actors at the sensors. The idea is to use concepts from Kalman filtering to reduce to the minimum the amount of information that is communicated by the actors to the sensors. In this section, we describe how the actors can estimate their position based on filtering of measurements. In Section 3.2.2.3, we explain how the measurements taken at the actor can be used, once communicated to the sensors, to estimate and predict the position of the actors. We will show in Section 3.5.1 that, in typical application scenarios, the actors need to communicate their measured position only when they deviate consistently from their current trajectory.

We define $\hat{\mathbf{x}}_\mathbf{i}^{\mathbf{k}-}$ as the *a priori* estimate at step k of the state of actor i, i.e., the estimate of the state given knowledge of the measurements up to step $k-1$. Formally, $\hat{\mathbf{x}}_\mathbf{i}^{\mathbf{k}-} = E\{\mathbf{x}_\mathbf{i}^\mathbf{k} | \mathbf{z}_\mathbf{i}^\mathbf{1}, .., \mathbf{z}_\mathbf{i}^{\mathbf{k-1}}\}$. Similarly, we define $\hat{\mathbf{x}}_\mathbf{i}^\mathbf{k}$ to be the *a posteriori* estimate at step k of the state of actor i, i.e., the estimate of the state given knowledge of the measurements up to step k. Formally, $\hat{\mathbf{x}}_\mathbf{i}^\mathbf{k} = E\{\mathbf{x}_\mathbf{i}^\mathbf{k} | \mathbf{z}_\mathbf{i}^\mathbf{1}, .., \mathbf{z}_\mathbf{i}^\mathbf{k}\}$. We can then define *a priori* and *a posteriori* estimate errors as $\mathbf{e}_\mathbf{i}^{\mathbf{k}-} = \mathbf{x}_\mathbf{i}^\mathbf{k} - \hat{\mathbf{x}}_\mathbf{i}^{\mathbf{k}-}$ and $\mathbf{e}_\mathbf{i}^\mathbf{k} = \mathbf{x}_\mathbf{i}^\mathbf{k} - \hat{\mathbf{x}}_\mathbf{i}^\mathbf{k}$, respectively. The *a priori* estimate

error covariance at step k can then be expressed as

$$\mathbf{P_i^{k-}} = E\{\mathbf{e_i^{k-}}\mathbf{e_i^{k-T}}\}, \tag{33}$$

while the *a posteriori* estimate error covariance at step k is

$$\mathbf{P_i^{k}} = E\{\mathbf{e_i^{k}}\mathbf{e_i^{kT}}\}. \tag{34}$$

The Kalman filtering equations can then be used at the actor to estimate its position, given the measurement at step k. First, the *time update* equations are calculated as

$$\hat{\mathbf{x}}_\mathbf{i}^{\mathbf{k-}} = \mathbf{F}\hat{\mathbf{x}}_\mathbf{i}^{\mathbf{k-1}} + \mathbf{G}\mathbf{u}_\mathbf{i}^{\mathbf{k-1}}, \tag{35}$$

$$\mathbf{P_i^{k-}} = \mathbf{F}\mathbf{P_i^{k-1}}\mathbf{F^T} + \mathbf{Q}. \tag{36}$$

Equation (35) predicts the state at step k, given the estimate of the state and the known control input at step $k-1$, thus projecting the state ahead. Equation (36) projects the covariance matrix ahead.

Then, after actor i measures its new position $\mathbf{z_i^k}$, the *measurement update equations* are used to correct the prediction based on the new measurement:

$$\mathbf{K_i^k} = \mathbf{P_i^{k-}}\mathbf{H^T}(\mathbf{H}\mathbf{P_i^{k-}}\mathbf{H^T} + \mathbf{R})^{-1} \tag{37}$$

$$\hat{\mathbf{x}}_\mathbf{i}^{\mathbf{k}} = \hat{\mathbf{x}}_\mathbf{i}^{\mathbf{k-}} + \mathbf{K_i^k}(\mathbf{z_i^k} - \mathbf{H}\hat{\mathbf{x}}_\mathbf{i}^{\mathbf{k-}}) \tag{38}$$

$$\mathbf{P_i^k} = (\mathbf{I} - \mathbf{K_i^k}\mathbf{H})\mathbf{P_i^{k-}} \tag{39}$$

In particular, (37) updates the value of the Kalman gain $\mathbf{K_i^k}$. Equation (38) calculates the *a posteriori* estimate $\hat{\mathbf{x}}_\mathbf{i}^{\mathbf{k}}$. Basically, (38) corrects the *a priori* estimate in (35) by adding a term that is proportional to the so-called *innovation*, i.e., $(\mathbf{z_i^k} - \mathbf{H}\hat{\mathbf{x}}_\mathbf{i}^{\mathbf{k-}})$. The innovation represents the difference between the *a priori* estimate of the two-dimensional position of the actor $\mathbf{H}\hat{\mathbf{x}}_\mathbf{i}^{\mathbf{k-}}$ and the measured position $\mathbf{z_i^k}$. It can be thus seen as the prediction error

at step k. This term is used, as explained later in this section, to limit the frequency of location updates of the actor. Finally, (39) calculates the new *a posteriori* estimate error covariance.

3.2.2.3 Sensor-side Position Prediction

The position of actor i can be estimated and predicted at the sensors in its Voronoi cell, based on the measurements $\mathbf{z_i^k}$ taken at the actor and broadcasted by the actor in its Voronoi cell. At step k, each sensor s in i's Voronoi cell updates the state (that represents position and velocity of the actor) based on the equations

$$\mathbf{\hat{x}_{i,s}^{k-}} = \mathbf{F}\mathbf{\hat{x}_{i,s}^{k-1}}, \tag{40}$$

$$\mathbf{P_{i,s}^{k-}} = \mathbf{F}\mathbf{P_{i,s}^{k-1}}\mathbf{F^T} + \mathbf{Q}. \tag{41}$$

Equation (40) is similar to (35), and describes how sensor s predicts the state of actor i before receiving the measurement. Note that the control input $\mathbf{u_i^{k-1}}$ is not known at the sensor. Similarly, (41) describes how sensor s projects the covariance matrix ahead. After receiving the measurement from actor $\mathbf{z_i^k}$, sensor s updates the Kalman Gain, and corrects the state estimate and covariance matrix according to the measurement, i.e.,

$$\mathbf{K_{i,s}^k} = \mathbf{P_{i,s}^{k-}}\mathbf{H^T}(\mathbf{H}\mathbf{P_{i,s}^{k-}}\mathbf{H^T} + \mathbf{R})^{-1} \tag{42}$$

$$\mathbf{\hat{x}_{i,s}^k} = \mathbf{\hat{x}_{i,s}^{k-}} + \mathbf{K_{i,s}^k}(\mathbf{z_{i,s}^k} - \mathbf{H}\mathbf{\hat{x}_{i,s}^{k-}}) \tag{43}$$

$$\mathbf{P_{i,s}^k} = (\mathbf{I} - \mathbf{K_{i,s}^k}\mathbf{H})\mathbf{P_{i,s}^{k-}}. \tag{44}$$

Note that the complexity of the above computations is very limited due to the small size of the state space of the problem. Moreover, the processing cost for sensors is much lower than the communication cost [85]. This is justified by experimental results on sensor

network prototypes such as [85], where the energy necessary to transmit 1 kbit is shown to be equivalent to the energy necessary to execute 300,000 processor instructions.

The actor will decide whether to send the updated measurement of its position based on the following.

$$\begin{cases} if\ (\mathbf{z_i^k} - \mathbf{H}\hat{\mathbf{x}}_{\mathbf{i,s}}^{\mathbf{k-}}) > e_{max} & send\ \mathbf{z_i^k} \\ \\ otherwise & do\ nothing \end{cases} \tag{45}$$

Hence, sensors are able to predict the new position of actors based on the past history of measurements broadcasted by the actor itself. At each step k, actor i broadcasts the new measurement $\mathbf{z_i^k}$ if and only if a sensor s in its cell, which has received the previous updates, is not able to predict the position of the actor within a maximum error e_{max}. If sensor s does not receive a location update at step k, it assumes $\mathbf{z_i^k} = \mathbf{H}\hat{\mathbf{x}}_{\mathbf{(i,s)}}^{\mathbf{k-}}$, i.e., the predicted position coincides with the actual new position of the actor. Based on this, it updates its estimate of the state for actor i as in (42-44).

3.3 Sensor-Actor Communication

In Chapter 2, we proposed a new notion of reliability that accounts for the percentage of packets generated by the sensors in the event area that are received within a pre-defined latency bound. The *event reliability* r perceived by an actor is the ratio of *reliable* data packets over all the packets received in a decision interval [1], where a packet is considered reliable if it is received within a given latency bound. The *event reliability threshold* r_{th} is the minimum event reliability required by the application. Unlike other more conventional notions of reliability, this definition is related to the timely delivery of data packets from sources to actors, and is calculated at the network layer. Note that we do not aim at devising

[1] Whenever a packet is dropped by an intermediate sensor, either because it violates the latency bound constraint or because of network or channel impairments, the actor is notified so that the lost packet can be taken into account in the computation of the reliability.

a solution that guarantees full reliability or that provides hard real-time guarantees on data delivery. Rather, the objective is to trade off energy consumption for latency when data has to be delivered within a given time bound B with a given reliability r_{th}. The solution presented in 2, based on similar premises, is however not suitable for mobile actors, as the convergence of the distributed protocol to an energy-efficient and latency compliant solution is too slow as compared to the dynamics encountered in networks with mobile actors. Therefore, when the traffic generated in the event area is low or moderate, we adjust the end-to-end delay by increasing the forwarding range with respect to the energy-efficient forwarding range, as described in Section 3.3.1. In case of congestion at a recipient actor, we re-route part of the traffic to another, less congested, actor.

3.3.1 Power-controlled Energy-delay Adjustment

Previous work on geographical routing considered primarily greedy forwarding[2] whereby a packet is forwarded to the closest node to the destination. However, this usually entails selecting links that connect the forwarding node to neighbors that reside close to the border of the transmission range. When a realistic physical layer is considered, such links are likely to be unstable and prone to high packet error rates. Hence, [93][98] propose enhanced flavors of greedy forwarding that avoid using those links. However, the objective is still to maximize the advance towards the destination, while we propose to forward packets on energy-efficient links, by trading off advancement at every single hop to minimize the energy consumption, unless a higher advancement is needed to increase the reliability. Moreover, as in [93][98], we remove the *unit disk graph* assumption relied on by most routing research, and consider a more accurate connectivity model. Hence, we first derive the energy-efficient forwarding distance in the presence of a fast fading channel. Then, we propose a mechanism to decrease the end-to-end delay by increasing the transmit power.

[2]Greedy forwarding has been enhanced in [21] by introducing face/perimeter routing techniques to route packets around the void area to reach the destination. This techniques can be applied to the mechanism proposed in this chapter in low-density or concave areas.

We model the fast fading channel with a Rayleigh statistics. Rayleigh fading occurs when there are multiple indirect paths between transmitter and receiver, with no distinct dominant path. Rayleigh fading, which is often used to model the effects of mobility in indoor or dense outdoor environments, can be considered a worst-case scenario, as in this extreme situation there is no clear desired signal. Furthermore, we assume the so-called *block fading model*, i.e., the attenuation due to fading remains constant during a packet transmission, but it is uncorrelated among subsequent transmission events. The effect of slow fading channels, which are usually modeled as log-normal envelopes, is left for future work.

Let us refer to the communication between v_i (forwarder) and v_j. If we denote their distance by d_{ij}, the probability $\mathcal{P}_{ij}^s$ that node v_j will receive a packet transmitted by v_i can be expressed as

$$\mathcal{P}_{ij}^s = \Pr\left\{\frac{P_{ij}^t \cdot f}{d_{ij}^\alpha} \geq \Gamma\right\}, \tag{46}$$

where P_{ij}^t is the power transmitted at v_i, Γ is a technology-dependent parameter representing the receiver threshold, and f is a unit-mean Rayleigh distributed r.v. that models fast fading for a given packet. Hence, we can write

$$\mathcal{P}_{ij}^s = \Pr\left\{f \geq \frac{\Gamma d_{ij}^\alpha}{P_{ij}^t}\right\} = \int_{\frac{\Gamma d_{ij}^\alpha}{P_{ij}^t}}^{+\infty} p_f(f)df = e^{-\frac{\pi}{4}\left(\frac{\Gamma d_{ij}^\alpha}{P_{ij}^t}\right)^2}. \tag{47}$$

The transmit power P_{ij}^t is related to the distance-dependent energy consumption through the transmit rate b as $P_{ij}^t = E_{\text{amp}} \cdot d_{ij}^\alpha \cdot b$. We can interpret $E_{\text{amp}} \cdot d_{ij}^\alpha \cdot b$ as the power necessary to transmit a packet over a distance d_{ij}, given a target packet error rate. The expression can be generalized by including a term that allows adjusting the desired packet error rate as follows

$$P_{ij}^t = (E_{\text{marg}} + E_{\text{amp}}) \cdot d_{ij}^\alpha \cdot b. \tag{48}$$

A higher value for E_{marg} leads to a higher energy consumption, and at the same time increases the probability of successful reception at the receiver, thus decreasing the expected

number of retransmissions. This is expressed by

$$\mathcal{N}_{ij}^{R}(d, E_{\text{marg}}) = \frac{1}{\mathcal{P}_{ij}^{s}} = e^{-\frac{\pi}{4}\left[\frac{\Gamma}{(E_{\text{marg}}+E_{\text{amp}})\cdot b}\right]^2}. \tag{49}$$

Now, consider a node v_i forwarding a packet towards a destination actor a_k at distance D. We consider the link metric $E = 2E_{\text{elec}} + E_{\text{amp}}d^{\alpha}$, where α is the path loss propagation exponent ($2 \leq \alpha \leq 5$), E_{amp} is a constant $[J/(bits \cdot m^{\alpha})]$, and E_{elec} is the energy needed by the transceiver circuitry to transmit or receive one bit $[J/bits]$. The end-to-end energy consumption can then be expressed as

$$E_{\text{e-e}} = \sum_{(i,j)\in\mathcal{P}(v_i,a_k)} \left(\frac{P_{ij}^{t}}{r} + 2E_{\text{elec}}\right), \tag{50}$$

where $\mathcal{P}(v_i, a_k)$ represents the path between v_i and a_k. Ideally, the end-to-end energy consumption is minimized when data are forwarded on a set of nodes located on the line connecting the source and the destination, equally spaced with internode distance d^{opt}. By plugging (48) in (50), and by considering retransmissions, we obtain

$$E_{\text{e-e}}^{\min} = \min_{d,E_{\text{marg}}} \left\{ \frac{D}{d_{ij}}[2E_{\text{elec}} + (E_{\text{marg}} + E_{\text{amp}})d_{ij}^{\alpha}] \cdot \mathcal{N}_{ij}^{R} \right\}$$

where $\mathcal{N}_{ij}^{R}$ is given by (49). The values for (d, E_{marg}) that minimize the above expression can be found by solving the nonlinear system $\nabla E_{\text{e-e}} = \mathbf{0}$, i.e., $[\frac{\partial E_{\text{e-e}}}{\partial d}, \frac{\partial E_{\text{e-e}}}{\partial E_{\text{marg}}}] = [0, 0]$, to find the stationary points of the function. A sufficient condition for a stationary point to be a a minimum is that the Hessian $\nabla^2 E_{\text{e-e}}$ calculated at the stationary point is positive definite. Note that the *optimal forwarding distance* d^{opt} is independent of D, i.e., the distance between the forwarding node and the intended destination. The expression can be interpreted as the optimal trade-off between distance-independent and distance-dependent energy consumption, and lends itself well to the development of localized forwarding rules. In case of ideal channel, and with $E_{\text{marg}} = 0$, (51) is minimized when $d^{\text{opt}} = \sqrt[\alpha]{\frac{2\cdot E_{\text{elec}}}{E_{\text{amp}}(\alpha-1)}}$. With the parameters given in [51], i.e., $E_{\text{elec}} = 50nJ/bit$, $E_{\text{amp}} = 100pJ/bit/m^{\alpha}$, $\alpha = 2.5$, the optimal forwarding distance for an ideal channel is $d^{\text{opt}} = 13.47m$. Solving (51) yields $d^{\text{opt}} = 8.00m$ and $E_{\text{marg}}^{\text{opt}} = 86pJ/bit/m^{\alpha}$, i.e., $E_{\text{marg}}^{\text{opt}} \approx E_{\text{amp}}$. Hence, as expected

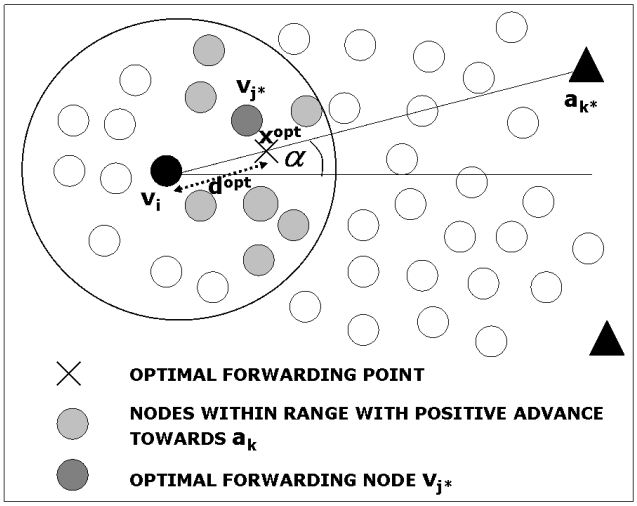


Figure 22: Illustration of Optimal Forwarding.

the optimal forwarding distance on a Rayleigh fading channel is lower than with an ideal channel, and a higher transmitting power is needed. It can be concluded that the energy-optimal path is obtained by forwarding the packet to a node that is located d^{opt} meters away on the line connecting the forwarding node and the destination. We refer to this point on the 2D plane as the *optimal forwarding point* $\mathbf{x}^{\text{opt}} = [x^{\text{opt}}, y^{\text{opt}}]$. A practical forwarding rule should intuitively select the next hop with minimal distance from this point. However, it can be demonstrated that for values of α (path loss exponent) higher than 3.5, the expected energy consumption increases excessively when the next hop is closer than the optimal forwarding point to the destination. Hence, in this case, the next hop is selected as the closest node to the optimal forwarding point, among those that are not closer to the destination than the optimal forwarding point.

Algorithm 5 Optimal forwarding for node v_i

Given:
v_i, the set of neighbors of v_i $\mathcal{N}(v_i)$, and the set of actors $\mathcal{A}$:
$k^* = argmin_k(\delta(v_i, a_k)), a_k \in \mathcal{A}$
$\alpha = \tan^{-1} \frac{(y_{k^*} - y_i)}{(x_{k^*} - x_i)}$
$x^{\text{opt}} = x_i + d^{\text{opt}} \cdot \cos\alpha$
$y^{\text{opt}} = y_i + d^{\text{opt}} \cdot \sin\alpha$
$j^* = argmin_j(\delta([x^{\text{opt}}, y^{\text{opt}}], v_j)), v_j \in \mathcal{N}(v_i) \cap \mathcal{P}(v_i, a_k)$

According to Algorithm 5, each sensor node v_i selects its closest actor a_k^* as its destination (where $\delta()$ indicates Euclidean distance). Then, it calculates the angle α formed by the ideal line connecting itself and the destination actor, and a reference direction, as shown in Fig. 22. It then calculates the optimal forwarding point by projecting d^{opt} in the direction of a_k^*. The optimal forwarding point $\mathbf{x}^{\text{opt}}$ in the figure is at distance d^{opt} from v_i on the line towards a_k^*. Finally, the next hop v_{j*} is selected as the closest neighbor with positive advance to the optimal forwarding point. Note that $\mathcal{P}(v_i, a_k)$ represents the set of nodes with positive advance towards a_k with respect to v_i.

It can be shown [19] that, given a Poisson deployment, the distance d_{rand} between a point in the plane and the node closest to that point is Rayleigh distributed, with mean $E\{d_{\text{rand}}\} = \frac{1}{2\sqrt{\varrho}}$, where ϱ is the deployment density. The above expression turns out useful to estimate how far we can expect the actual forwarding node to be from the optimal forwarding point calculated in Algorithm 5. Since the closest node is expected to be $\frac{1}{2\sqrt{\varrho}}$ meters far from the optimal forwarding point, we can conclude that the forwarding node selected by Algorithm 5 gets closer to the optimal forwarding point as the density increases. For example, if we deploy 100 sensors in a square $100mx100m$ terrain, we can expect a node to be $\frac{1}{2\sqrt{1/100}} = 5m$ far from the optimal forwarding point.

Algorithm 6 describes how to control the reliability by means of actor feedback messages. We adopt a conservative approach. When an event occurs, all sensors start transmitting with the maximum forwarding range. Then, according to the actor feedback on the observed reliability, sensors may decrease their forwarding range until either the reliability is close to the required event reliability threshold r_{th}, or until the optimal forwarding range is reached. Transmitting closer than the optimal forwarding range, as will be shown in Section 3.5.1, leads to high delay and high energy consumption, and is thus avoided. When the observed reliability is low even with the longest forwarding ranges, the actor initiates procedures for network layer congestion control, as explained in Section 3.3.2.

Algorithm 6 Reliability control

$d = d^{max}$
Calculate reliability r_i
while ($r_i > r_{th} - \epsilon$) and ($d > d^{opt}$) **do**
  $d = d - \Delta d$
end while
while $r_i \leq r_{th}$ **do**
  Calculate optimal actor a_{k*}
  Send $virtual\ position$ $\mathbf{x}_{\mathbf{k*}}^{\mathbf{virt}}$
end while

3.3.2 Network Layer Actor-driven Congestion Control

In several application scenarios high sampling rates at the sensors, large event areas, or dense deployment may lead to high contention and consequent collisions at the MAC layer, and ultimately to decreased reliability. In classical network theory, these situations are usually handled by decreasing the data rate by means of congestion control algorithms at the transport layer. However, although congestion control mechanisms have been devised for sensor networks [12], these usually rely on spatial correlation among sampled data and assume that the sampling rate at the sensors can be changed. Nevertheless, the peculiar characteristics of WSANs, and in particular the equivalence of different actors as recipients for sensor data, allow devising procedures to relieve congested actors from excessive traffic burden by deviating traffic towards other idle actors. Indeed, the objective of such a procedure is to trade off energy consumption, by reaching a suboptimal actor, for increased reliability. To do so, there is a need to develop a mechanism to allow congested actors to detect situations of congestion, and to identify suitable alternate actors to re-route traffic to, to notify sensors that a different actor needs to receive their data. In this section, we propose a mechanism to take countermeasures at the network layer.

We propose to detect congestion at the actor receiving data and redirecting traffic to other, less congested, actors. We consider the notion of reliability from [75], as recalled at the beginning of this section. Whenever an actor a_i detects very low reliability, caused by excessive delays and packet drops, it selects another actor to re-route the traffic from half

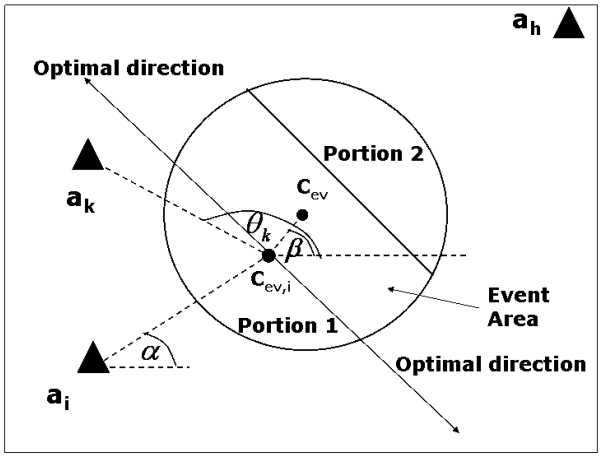


Figure 23: Calculation of the directivity factor δ_i.

of the sensors in its Voronoi cell to that actor. Each actor a_k is assigned by a_i a weight w_k, which measures its suitability to become a recipient for the traffic generated in the portion of the event area which a_i is receiving data from. The weight w_k, which is low for better-suited actors, is calculated as the weighted sum of three factors, $w_k = \frac{c_\eta \eta_k + c_\delta \delta_k + c_\Delta \Delta_k}{c_\eta + c_\delta + c_\Delta}$, with weights c_η, c_δ, c_Δ. As a design choice, we set $c_\eta \geq c_\delta \geq c_\Delta$.

1) *Congestion factor* η_k, $0 \leq \eta_k \leq 1$. This normalized value reflects the reliability observed at actor a_k, i.e., $\eta_k = 1$ if $r < r_{th} - \epsilon$, it monotonically decreases as $r - r_{th}$ increases, and $\eta_k = 0$ for actors that are not receiving traffic. Here, ϵ represents a suitable margin on the reliability to avoid instability.

2) *Directivity factor* δ_k, that reflects the relative angular position of actor a_k with respect to actor a_i and the center of the event area.

Let us refer to Fig. 23, which illustrates the situation where an actor a_i is receiving data from part of the event area. We indicate the center of the event area as C_{ev}, which represents the weighted sum of the positions of the sensors. The center of the portion of the event area that resides in a_i's cell is referred to as $C_{ev,i}$. In the example given in Fig. 23, the event area is divided into two parts, and another actor receives data from the second portion of the event area. However, the proposed procedure to calculate the directivity factor holds

in the general case where the event area is divided among multiple actors, given that the center of the global event C_{ev} has been collaboratively reconstructed by the participating actors. The idea is to give higher weights to actors that reside in the same direction of a_i with respect to $C_{ev,i}$, as this would cause increased traffic in the direction of a_i; or in the direction of C_{ev} with respect to $C_{ev,i}$, as this would increase traffic in the event area. Rather, the directivity factor should be maximum for those actors that are away from these two directions (optimal directions in Fig. 23). The angles α, β, and θ_k describe the relative angular positions of $C_{ev,i}$ and a_i, C_{ev}, and a_k, respectively. After some derivations, the directivity factor for actor a_k can be calculated as follows

$$\delta_k = \begin{cases} \frac{2\theta_k+(\pi-\beta-\alpha)}{(\pi+\beta-\alpha)} & 0 \leq \theta_k \leq \beta \\ \frac{|2\theta_k-(\pi+\beta+\alpha)|}{(\pi+\alpha-\beta)} & \beta \leq \theta_k \leq \pi+\alpha \\ \frac{|2\theta_k-(3\pi+\alpha+\beta)|}{(\pi+\beta-\alpha)} & \pi+\alpha \leq \theta_k \leq 2\pi. \end{cases} \tag{51}$$

3) *Distance factor* Δ_k, which is the distance of the actor from the center of the event $C_{ev,i}$ normalized to the diameter of the monitored area, i.e., $\Delta_k = 1$ when the distance is maximal.

A congested actor a_i selects the optimal actor a_{k*} with minimum weight w_{k*}. Then, actor a_i calculates and advertises a new *virtual position* $\mathbf{x}_{\mathbf{k*}}^{\mathbf{virt}}$ for a_{k*} to the sensors in its Voronoi cell. The virtual position is forced to be on the line connecting the real position of the actor $\mathbf{x}_{\mathbf{k*}}$ and the center of the event area $C_{ev,i}$, and corresponds to the point such that half of the sensors in $C_{ev,i}$ are closer to a_i, while the other half is closer to a_{k*}. Each sensor will select its recipient actor, using for actor a_{k*} the virtual position $\mathbf{x}_{\mathbf{k*}}^{\mathbf{virt}}$, while the real position $\mathbf{x}_{\mathbf{k*}}$ is still used to perform the actual forwarding function. The concept of virtual position allows to optimally partition the sensors in such a way that only those that are closer to a_{k*} redirect their traffic to it, and provides a compact way to notify the sensors. The procedure is applied recursively by actors that are still congested after splitting the

traffic in two.

Algorithm 7 describes the procedure run by actor a_i to calculate the virtual position for actor a_k. The symbols $\mathbf{x_i}$ and $\mathbf{x_{k*}}$ refer to the position of actors a_i and a_{k*}, while $\mathcal{S}_i$ refers to the set of sources that reside in the portion of the event area closer to a_i.

Algorithm 7 Calculate virtual position for actor a_{k*}

$\mathbf{x}_{\mathbf{k*}}^{\mathbf{virt}} = \mathbf{x_{k*}}$
$\mathbf{x}_{\mathbf{k}}^{\mathbf{last}} = \mathbf{x_{k*}}$
N_i = Calculate sensors in $\mathcal{S}_i$ closer to $\mathbf{x_i}$
N_{k*}^{virt} = Calculate sensors in $\mathcal{S}_i$ closer to $\mathbf{x}_{\mathbf{k*}}^{\mathbf{virt}}$
while $|N_i - N_{k*}^{virt}| > 1$ **do**
  if $N_i > N_{k*}^{virt}$ **then**
    $\mathbf{x}_{\mathbf{k}}^{\mathbf{last}} = \mathbf{x}_{\mathbf{k*}}^{\mathbf{virt}}$
    $\mathbf{x}_{\mathbf{k*}}^{\mathbf{virt}} = (\mathbf{x}_{\mathbf{k*}}^{\mathbf{virt}} + \mathbf{C_{ev,i}})/2$
  else
    $\mathbf{x}_{\mathbf{k*}}^{\mathbf{virt}} = (\mathbf{x}_{\mathbf{k*}}^{\mathbf{virt}} + \mathbf{x}_{\mathbf{k}}^{\mathbf{last}})/2$
  end if
  N_i = Calculate sensors in $\mathcal{S}_i$ closer to $\mathbf{x_i}$
  N_{k*}^{virt} = Calculate sensors in $\mathcal{S}_i$ closer to $\mathbf{x}_{\mathbf{k*}}^{\mathbf{virt}}$
end while

3.4 Actor-Actor Coordination

In this section, we formulate the multi-actor task allocation problem, whose objective is to coordinate the mobility. In particular, it selects the best actor(s) to form the *actor team*, and to control their motion toward the action area. Our previous work [75] assumes that static actors are only able to act within a circular area defined by their action range. Hence, it is not suitable for WSANs with mobile actors. Moreover, in [75] reallocation of resources to face multiple events is not considered. Here, we introduce a more general framework and remove these assumptions.

The position of the sensors that generate readings defines the *event area*. The *action area* represents the area where the actors should act, and is identified by processing the event data. In general, the event and the action areas may be different, although they may coincide in several applications.

We consider a multi-event scenario where multiple events may give rise to event/action areas partially overlapped in space and/or time, and an event may occur before the actions associated with previous events have been successfully completed. The proposed allocation problem presents analogies with the class of so-called *Multi-Robot Task Allocation* (MRTA) problems encountered in robotics [44]. We are concerned with methods for *intentional cooperation*, i.e., mobile actors cooperate explicitly through task-related communication and negotiation, and coordinate their motion to efficiently act on the action areas, based on the characteristics of the reconstructed events. Other approaches to cooperation, such as minimalist or emergent approaches [44], where individual actors coordinate their actions without explicit negotiation or allocation of tasks, are out of the scope of this chapter.

According to the event features collected from the event area, each occurring event ω in the *event space* Ω can be characterized by the tuple $\mathcal{E}^{(\omega)} = \{F^{(\omega)}, Pr^{(\omega)}, A^{(\omega)}, S^{(\omega)}, I^{(\omega)}, D^{(\omega)}\}$, where $F^{(\omega)}$ describes the *event type*, i.e., the class the event belongs to, $Pr^{(\omega)}$ the *priority*, $A^{(\omega)}[m^2]$ the *event area*, $S^{(\omega)}[m^s]$ and $I^{(\omega)}[J/m^2]$ the *scope* (the action area) and *intensity*, respectively, and $D^{(\omega)}[s]$ the *action completion bound*, i.e., the maximum allowed time from the instant when the event is sensed to the instant when the associated action needs to be completed. These characteristics, which define each occurring event, are distributively reconstructed by the actors that receive sensor information, and constitute inputs to the multi-actor task allocation problem. In particular, the multi-actor allocation problem consists of selecting a *team of actors* and their *velocity* to optimally divide the action workload, so as to minimize the energy required to complete the action, while respecting the *action completion bound*. Although actors are resource-rich nodes, the order of magnitude of the energy required for actions and for movements is higher than that required for communication. Hence, it is important to save action and movement energy to extend the lifetime of actors. We formulate the multi-actor allocation problem as a *Mixed Integer Non-Linear Program* (MINLP).

In the following, The objective is to find, for each occurring event $\omega \in \Omega$, the subset of

actors and their optimal velocities such a way to minimize the energy required to complete the action associated with the occurring event, under the constraint of meeting the action completion bound.

We define the problem according to the following assumptions: i) the energy to perform the action (action and movement energy) is orders of magnitude higher than the energy required for communication; ii) task reallocation is performed only if actions associated with higher priority events cannot be accomplished due to lack of resources.

We introduce the following notation:

- $l_a^f\ [W]$ is the *action power level* of actor a, when the event type $f \in \mathcal{F}^{(\omega)}$;
- $T_a^{\Omega,(\omega)}[s]$ is the time actor a needs to complete the action associated with event ω when a is part of an acting team;
- $E_a^{\Omega,(\omega)} = l_a^f \cdot T_a^{\Omega,(\omega)}[J]$ is the energy required by a to complete its task, given its action power level and action time;
- $d_a^{(\omega)}[m]$ is the distance between actor a and the center of the action area $S^{(\omega)}$, while $T_a^{M,(\omega)}[s]$ is the time needed by actor a to reach it;
- $E_a^{M,(\omega)} = [\beta {v_a^{(\omega)}}^{\gamma} + P_{min}^M] \cdot T_a^{M,(\omega)}[J]$ is the energy actor a requires to move at speed $v_a^{(\omega)}$ for $T_a^{M,(\omega)}$ seconds, where $P_{min}^M[W]$ is a velocity-independent term that accounts for dissipative effects;
- $\mathbf{X}^{(\omega)}$ is a binary vector whose element $[x_a^{(\omega)}]$ is equal to 1 iff actor a acts on the action area $S^{(\omega)}$ defined by event $\omega \in \Omega$;
- $\mathbf{V}^{(\omega)}$ is a vector whose element $[v_a^{(\omega)}]$ represents the velocity assigned to actor a;
- η_a^f is the *efficiency* of actor a acting on an event type $f \in \mathcal{F}^{(\omega)}$, i.e., the ratio between the effect produced by the action energy applied to the action area and the action energy itself;

- $E_a^{Av}[J]$ is the *available energy* of actor a evaluated at the instant when event ω occurs;
- $T^C[s]$ is the *coordination delay*, i.e., the time needed to process the event data, reconstruct the event itself, and select the team of actors by solving problem $\mathbf{P}_{All}^{(\omega)}$; note that the coordination delay does not depend on the event;
- $\mathcal{S}_A^I \in \mathcal{S}_A$ is the subset of actors in *IDLE* state when event ω occurs, i.e., actors that have not been assigned to act on action areas associated with previously occurred events;
- N_S^a is the total number of sources sending packets to actor a, while $\Psi(N_S^a)[J]$ is a *penalty function* weighting the choice of actor a, which is receiving data from N_S^a sources, to be part of an acting team. The penalty function monotonically increases as N_S^a increases.

We now formulate the multi-actor task allocation problem.

$\mathbf{P}_{All}^{(\omega)}$: **Multi-actor Task Allocation Problem**

$$Find: \qquad \mathbf{X}^{(\omega)} = [x_a^{(\omega)}],\ \mathbf{V}^{(\omega)} = [v_a^{(\omega)}]$$

$$Minimize: \quad \textstyle\sum_{a \in \mathcal{S}_a^I} x_a^{(\omega)} \cdot [E_a^{M,(\omega)} + E_a^{\Omega,(\omega)} + \Psi(N_S^a)]$$

$$Subject\ to:$$

$$E_a^{M,(\omega)} = [\beta {v_a^{(\omega)}}^{\gamma} + P_{min}^M] \cdot T_a^{M,(\omega)},\ \forall a \in \mathcal{S}_a^I; \tag{52}$$

$$T_a^{M,(\omega)} = \frac{d_a^{(\omega)}}{v_a^{(\omega)}},\ \forall a \in \mathcal{S}_a^I; \tag{53}$$

$$v_a^{min} \leq v_a^{(\omega)} \leq v_a^{max},\ \forall a \in \mathcal{S}_a^I; \tag{54}$$

$$E_a^{\Omega,(\omega)} = l_a^f \cdot T_a^{\Omega,(\omega)} \geq 0,\ \forall a \in \mathcal{S}_a^I,\ f \in \mathcal{F}^{(\omega)}; \tag{55}$$

$$\sum_{a \in \mathcal{S}_a^I} x_a^{(\omega)} \cdot \eta_a^f \cdot E_a^{\Omega,(\omega)} \geq S^{(\omega)} \cdot I^{(\omega)},\ f \in \mathcal{F}^{(\omega)}; \tag{56}$$

$$T_a^{M,(\omega)} + T_a^{\Omega,(\omega)} \leq D^{(\omega)} - T^C, \ \forall a \in \mathcal{S}_a^I; \tag{57}$$

$$E_a^{M,(\omega)} + E_a^{\Omega,(\omega)} \geq E_a^{Av}, \ \forall a \in \mathcal{S}_a^I; \tag{58}$$

$$\sum_{a \in \mathcal{S}_a^{F,(\omega)}} x_a^{(\omega)} \geq 1. \tag{59}$$

Constraint (52) defines the energy required for actor a to move to the action area defined by the occurring event, which is the product of the power needed to move and the time needed to reach the action area at a given velocity; this time is expressed as the ratio between the distance of the actor from the action area and the selected velocity, as expressed in (53). Constraint (54) bounds the velocity range for each actor. Constraint (55) defines the energy required for actor a to complete the action when it is part of an acting team. Constraint (56) assures that the selected team be able to complete the assigned task, given the characteristics of the actor composing the team, and the scope and intensity of the event. Constraint (57) limits the sum of the action completion time and the time required to move the actor team to be smaller than the action completion bound, discounted by the coordination delay. Constraint (58) guarantees a non-negative residual energy for each actor. Finally, constraint (59) ensures that at least one actor act on the advertised action area.

Event Preemption for Multi-actor Task Allocation: Algorithm 8 defines the event-preemption policy for multi-actor task allocation in the case where resources are insufficient to accomplish a high priority task. For the sake of simplicity, task reallocation is performed only if actions associated with higher priority events cannot be accomplished because of lack of resources, as it stated in the assumptions reported in this section. More specifically, if the task associated with event ω cannot be accomplished, given the resource already allocated to all active events (Ω_{Active}), i.e., if $\mathbf{P}_{All}^{(\omega)}/\Omega_{Active}$ is unfeasible, then Algorithm 8 proceeds with the preemption of all those ongoing tasks characterized by lower priorities, if any. The objective of this preemptive scheme is to reallocate useful resource to higher priority events that could not be successfully completed otherwise, while minimizing the

Algorithm 8 Event preemption for multi-actor task allocation

if $\mathbf{P}_{All}^{(\omega)}/\Omega_{Active} == FEASIBLE$ **then**
  $SOLVE\,(\mathbf{P}_{All}^{(\omega)}/\Omega_{Active})$
  $\Omega_{Active} \equiv \Omega_{Active} \cup \omega$
  $UPDATE\,(\mathcal{S}_A^I)$
else
  $\sigma_{min} = argmin_{\sigma \in \Omega_{Active}} Pr^{(\sigma)}$
  if $Pr^{(\omega)} > Pr^{(\sigma_{min})}$ **then**
    $\Omega'_{Active} \equiv \Omega_{Active} \setminus \sigma_{min}$
    $UPDATE\,(\mathcal{S}_A^I)$
    $SOLVE\,(\mathbf{P}_{All}^{(\omega)}/\Omega'_{Active})$
    $\Omega''_{Active} \equiv \Omega'_{Active} \cup \omega$
    $UPDATE\,(\mathcal{S}_A^I)$
    if $\mathbf{P}_{All}^{(\sigma_{min})}/\Omega''_{Active} == FEASIBLE$ **then**
      $SOLVE\,(\mathbf{P}_{All}^{(\sigma_{min})}/\Omega''_{Active})$
      $\Omega_{Active} \equiv \Omega''_{Active} \cup \sigma_{min}$
      $UPDATE\,(\mathcal{S}_A^I)$
    end if
  end if
end if

number of costly task reallocations.

Our actor-actor coordination mechanism includes an event-preemption policy for multi-actor task allocation for cases where resources are insufficient to accomplish a high priority task, which is omitted for lack of space.

3.5 Performance Results

In this section, we report results of simulative evaluation of the proposed mechanisms. In particular, Section 3.5.1 discusses our proposed algorithms for sensor-actor communication, while Section 3.5.2 evaluates our actor-actor coordination scheme.

3.5.1 Sensor-actor Communication

Performance results shown in this section are obtained with the sensor-actor simulator that we developed within the J-SIM framework [7]. First, we discuss results relevant to the prediction procedure described in Section 3.2. Actors move according to the model described

in Section 3.2.2. In the first set of simulations, each actor selects a target destination and moves at constant speed to reach it. The actor implements a proportional feedback controller that generates input commands to compensate for the process noise (random acceleration) by reestablishing the correct direction and speed. At each step, the actor measures its position (which is affected by measurement noise), filters the data, and decides whether an update needs to be sent.

In Figs. 24 and 25 we report the *failure rate* of the prediction procedure, with varying values for e_{max}, and for different values of the process noise (random acceleration). The failure rate is defined as the number of location updates sent over all measurements taken at the actor. Each figure reports results averaged over different simulation scenarios, with 95% confidence intervals. In Fig. 24 we report the failure rate with varying process noise, while in Fig. 25 we show the failure rate with varying measurement noise. In the range of values analyzed, which corresponds to realistic motion scenarios, it is shown that if it is possible to accept a localization error of $5\,\mathrm{m}$ for the actors, which is reasonable being around 10% of the transmission range, the prediction at the sensors allows the actor to avoid 75% and more location updates, with proportional energy savings at the sensors. In the second set of simulations, reported in Fig. 26, actors select several different destinations during each simulation, similarly to a (perturbed) Random Waypoint model. The failure rate is only slightly higher, which shows that the prediction procedure proposed is effective even when complicated movement patterns are in place, and shows good robustness properties against noise.

As far as sensor-actor communication is concerned, sensors implement the geographical forwarding algorithm described in Section 3.3. The MAC layer is based on CSMA/CA. At the physical layer, we implemented our power control procedure and set bandwidth and power consumption parameters similar to IEEE 802.15.4 compliant radios according to the Chipcon CC2420 datasheet. The monitored area is a $200\,\mathrm{m}$x$200\,\mathrm{m}$ square, with 200 randomly deployed sensors. The maximum transmission range of sensors is set to $40\,\mathrm{m}$, and

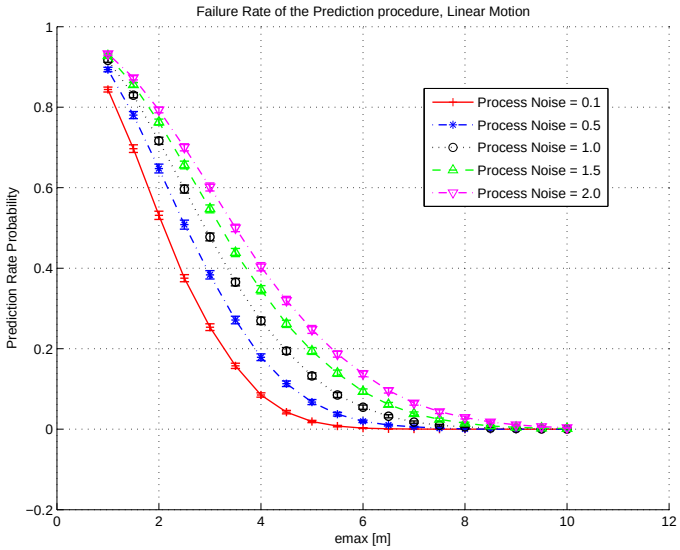


Figure 24: Failure rate of the prediction procedure, with linear motion, for different levels of process noise.

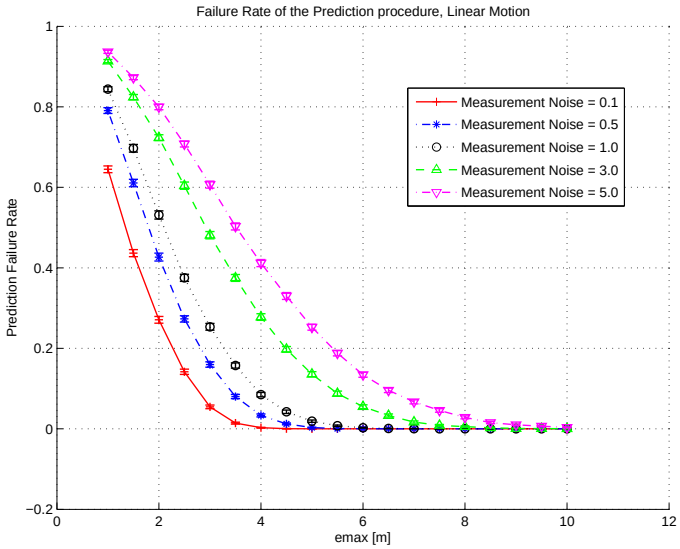


Figure 25: Failure rate of the prediction procedure, with linear motion, for different levels of measurement noise.

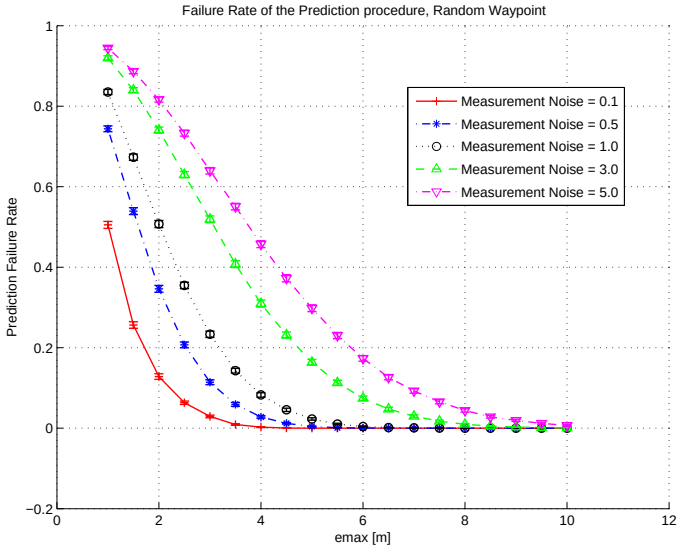


Figure 26: Failure rate of the prediction procedure, with random waypoint motion, for different levels of measurement noise.

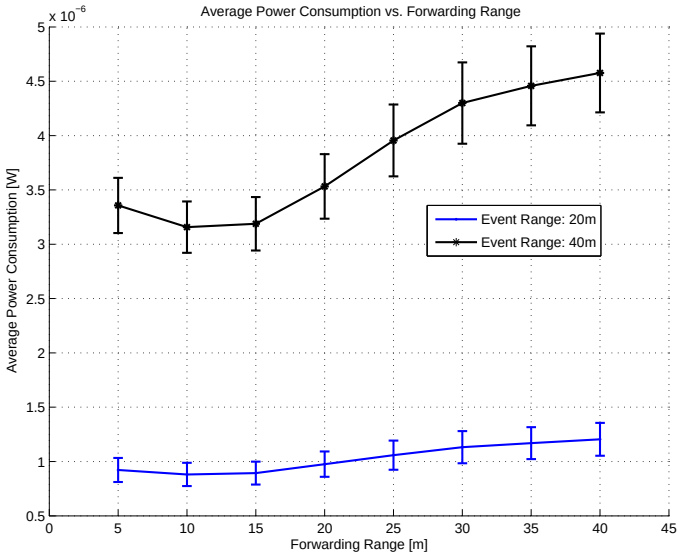


Figure 27: Average power consumption vs. forwarding range, low and moderate traffic.

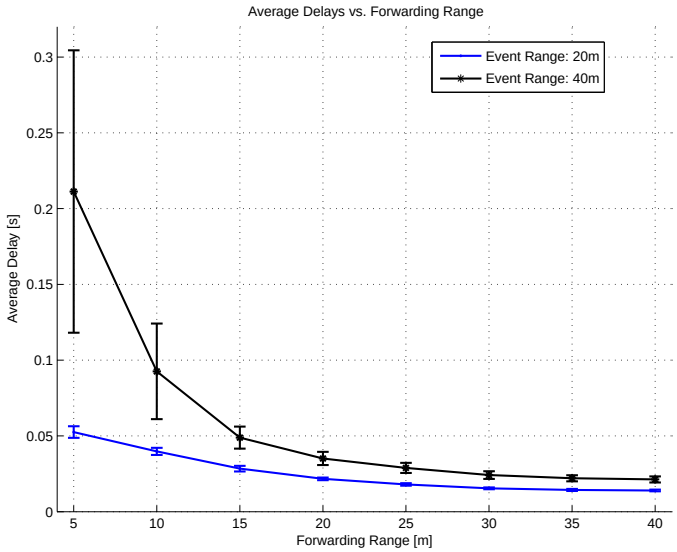


Figure 28: Delay vs. forwarding range, low and moderate traffic.

the bandwidth to $250\,\mathrm{kbit/s}$. Sensors send $56\,\mathrm{byte}$ long packets with a reporting rate of $1\,\mathrm{packet/s}$, and the size of the queues is set to 20 packets. We perform terminating simulations that last $400\,\mathrm{s}$, average over different random topologies, and show 95% confidence intervals. The simulation parameters are reported in Table 3.

Table 3: Simulation Parameters - Sensor-actor Communication

Parameter	**Value**
area	200x200 m
sensors	200
max tx range	40 m
bandwidth	250 kbit/s
packet size	56 bytes
queue	20 packets
reporting rate	1 packet/s
simulation time	400 s

In Figs. 27 and 28, we show a comparison of the average power consumption and delay, respectively, with increasing forwarding range. Sensors inside the event area report measurements to the actor. The *event area* is circular and centered at $(100, 100)\,\mathrm{m}$. The figures report simulation runs for the cases of low and moderate traffic, i.e, the *event range*

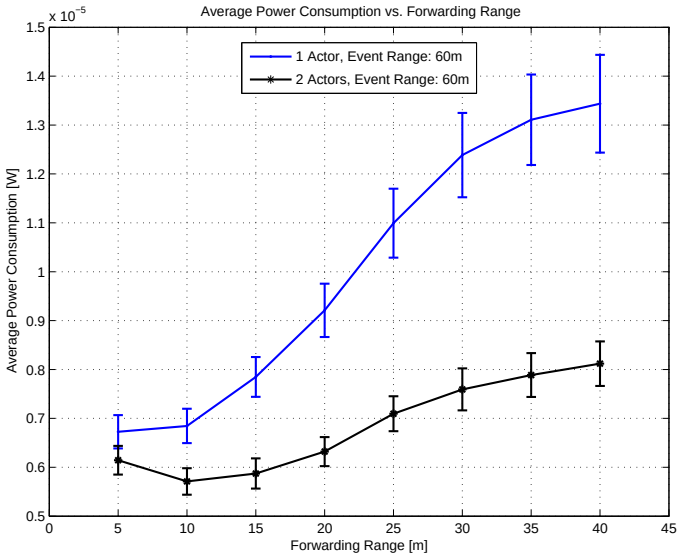


Figure 29: Average power consumption vs. forwarding range, high traffic.

is equal to $20\,\mathrm{m}$ and $40\,\mathrm{m}$ around the center, respectively. In the first case, on average 7 sensors reside in the event area, while in the second case there are around 25 sources. In Figs. 27 and 28 we show that in situations of low and moderate traffic, which are common in sensor networks, the end-to-end delay can be consistently decreased by increasing the forwarding range. This is an important trade-off that has not been thoroughly explored so far. Clearly, this is paid with increased power consumption with respect to the optimal values. Note that the power consumption obtained for very low values of the forwarding range, e.g., $5-10m$, are in general not much higher than values associated with the optimal forwarding range. The average power consumption seems to be flat, while it would be expected to decrease according to the model in Section 3.3.1. However, given the density of nodes $\varrho = 1/200$ in the considered scenario, as discussed in Section 3.3.1, the closest node to the optimal forwarding point is expected to be $1/2\sqrt{\varrho}\,m$ apart from the optimal forwarding point, i.e., around $7m$. This determines a practical upper bound on the number of hops of the end-to-end paths. For this reason, we do not see the power consumption increasing with very small forwarding range as in practice packets are forwarded to farther nodes.

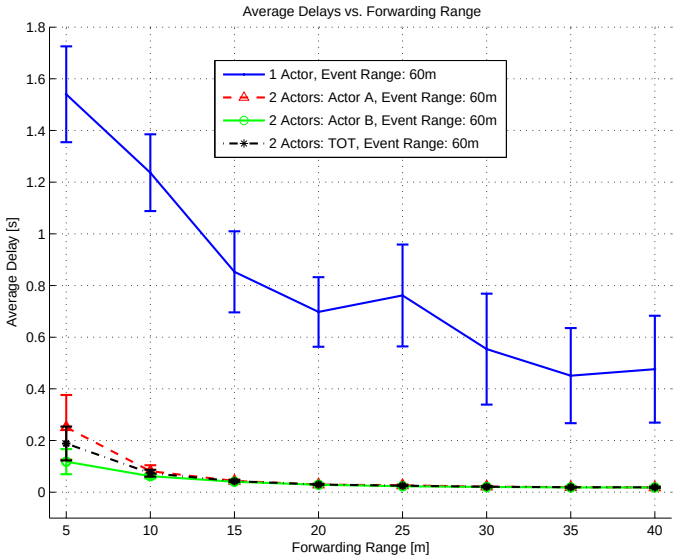


Figure 30: Delay vs. forwarding range, high traffic.

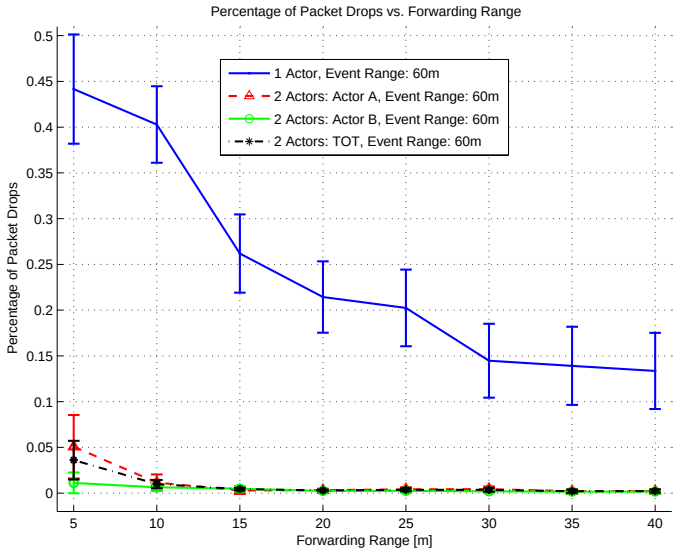


Figure 31: Packet drops vs. forwarding range, high traffic.

Figure 29 refers to a high traffic scenario. The event range is set to $60\,\mathrm{m}$, which corresponds to 57 sources on average. The event area lies completely in the Voronoi cell of a single actor. We compare energy consumption, delay, and packet drops when 1 or 2 actors receive the traffic generated in the event area, i.e., with or without the congestion control procedure devised in Section 3.3.2. We observe the following behavior. In the first case (no congestion control), the event area itself is congested, and a high percentage of packets are dropped (between 15% and 40%) (Fig. 31), while the end-to-end delays increase to about $1\,\mathrm{s}$ and are not easily controlled by changing the forwarding range (Fig. 30). Note that packets are dropped mostly in the event area due to multiple collisions at the MAC layer. Closer to the actor, the traffic is decreased due to earlier drops, and fewer nodes try to transmit simultaneously. Conversely, congestion can be dramatically decreased when the proposed congestion control procedure divides the event data between two actors. This is due to the fact that most of the congestion and packet drops occur in the event area, where many nodes try to transmit simultaneously, with the consequent drops due to simultaneous transmissions. This is dramatically improved when a second actor on the opposite side of the event area receives data, since traffic is diverted from the event area. The percentage of packets dropped is close to nil (see Fig. 31), delays are two orders of magnitude lower and can be regulated with power control (Fig. 30). Importantly, even though the second actor is farther (thus, in theory, suboptimal) from the event area, and although without congestion control packets are dropped early on their source-actor path, the power consumption is also decreased by the congestion control procedure, mostly due to reduced packet retransmissions at the MAC layer (Fig. 29).

3.5.2 Actor-actor Coordination

In this section, we discuss performance results for the multi-actor task allocation problem presented in Section 3.4. Actors are assumed to be randomly deployed in a $200\,\mathrm{m}\mathrm{x}200\,\mathrm{m}$ area, where events with intensity $I = 0.5\,\mathrm{J/m^2}$ and scope $S = \pi \cdot 4^2\,\mathrm{m}^2$ occur randomly

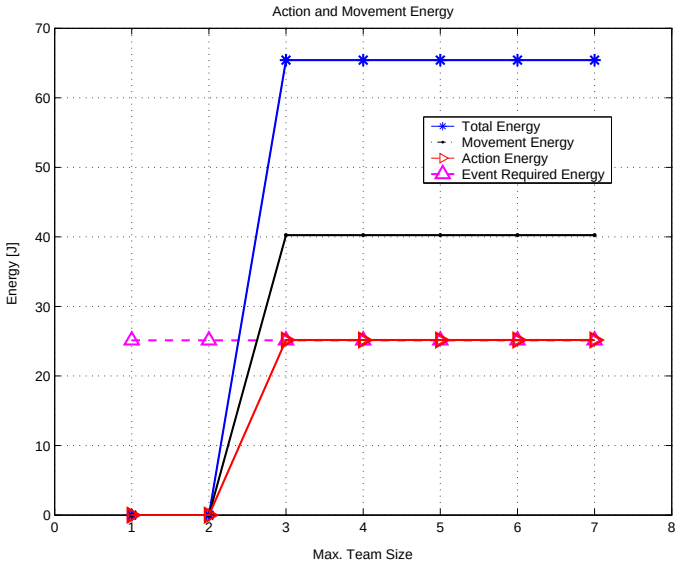


Figure 32: Energy consumption vs. maximum team size.

in the entire area.We set the action completion bound D and the coordination delay T^C to $15\,\mathrm{s}$ and $1\,\mathrm{s}$, respectively. We consider a scenario with homogeneous actors, with $\beta = 0.05\,\mathrm{W}/(\mathrm{m}/\mathrm{s})^{\gamma}$, $\gamma = 1.5$, $P^M_{min} = 1\,\mathrm{W}$, efficiency $\eta = 1$, action power $l = 1\,\mathrm{W}/\mathrm{m}^2$, and initial energy $E_0 = 1000\,\mathrm{J}$; moreover, the velocities range in the interval $[3, 12]\,\mathrm{m}/\mathrm{s}$.

Figures 32 and 33 report results from a set of simulations where we impose a limit on the maximum team size, i.e., the maximum number of actors taking part in an acting team, reported on the x axis, while in Fig. 34 the number of actors composing a team is forced to be fixed and equal to the team size, which is reported on the x axis. Interestingly, when the number of actors taking part into an acting team is optimized to minimize the overall energy expenditure, i.e., the sum of the movement energy E^M and the action energy E^{Ω}, at least 3 actors are needed to complete the action (see Fig. 32) and the total action time tends to be exactly the maximum allowed completion bound D, discounted by the coordination delay T^C (see Fig. 33). Problem $\mathbf{P}^{(\omega)}_{All}$ tends to minimize the number of involved actors, and to assign higher speed to those actors that are closer to the action area. This can be explained by considering that a fixed amount of power (P^M_{min}) is dissipated every time an actor needs to move, irrespective of its velocity. Conversely, when all the available actors

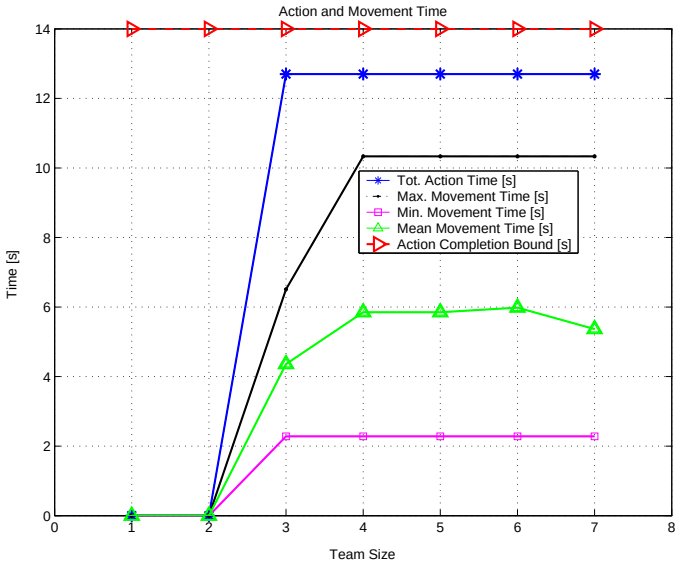


Figure 33: Delay vs. maximum team size.

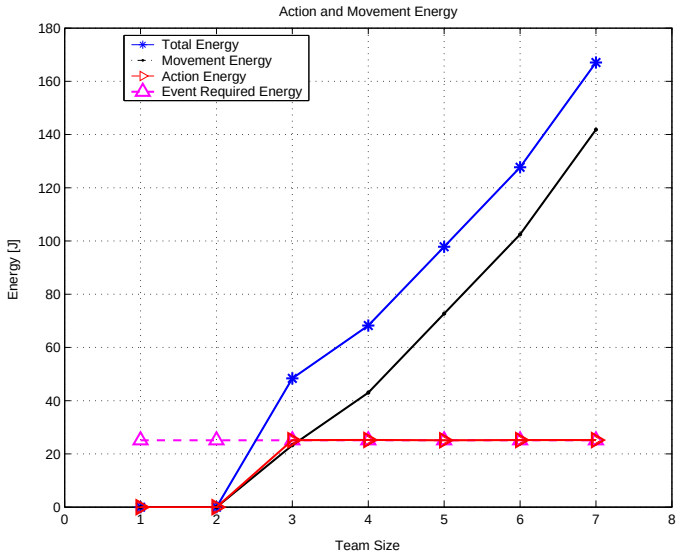


Figure 34: Energy consumption vs. team size.

are forced to be part of a team, the action time can be reduced at the expense of energy consumption, as reported in Fig. 34.

CHAPTER IV

CHALLENGES FOR MULTIMEDIA COMMUNICATIONS IN WIRELESS SENSOR AND ACTOR NETWORKS

4.1 Preliminaries

So far, we have discussed communication and coordination challenges in wireless sensor and actor networks in particular in data-centric scenarios. In this chapter, we introduce the main challenges for the delivery of multimedia content. In fact, we believe that the availability of inexpensive hardware such as CMOS cameras and microphones that are able to ubiquitously capture multimedia content from the environment will foster the development of Wireless Multimedia Sensor and Actor Networks (WMSANs) [49][77], i.e., networks of wirelessly interconnected devices that will allow retrieving video and audio streams, still images, and scalar sensor data, and perform actions in real time based on the multimedia content gathered. With rapid improvements and miniaturization in hardware, a single sensor device can be equipped with audio and visual information collection modules. As an example, the Cyclops image capturing and inference module [88], is designed for extremely light-weight imaging and can be interfaced with a host mote such as Crossbow's MICA2 [4] or MICAz [5]. In addition to the ability to retrieve multimedia data, WMSANs will also be able to store, process in real-time, correlate and fuse multimedia data originated from heterogeneous sources, and finally perform actions based on processing of the information gathered.

Wireless multimedia sensor and actor networks will not only enhance existing sensor network applications such as tracking, home automation, and environmental monitoring, but they will also enable several new applications such as:

- **Multimedia Surveillance Sensor Networks.** Wireless video sensor networks will

be composed of interconnected, battery-powered miniature video cameras, each packaged with a low-power wireless transceiver that is capable of processing, sending, and receiving data. Video and audio sensors will be used to enhance and complement existing surveillance systems against crime and terrorist attacks. Large scale networks of video sensors can extend the ability of law enforcement agencies to monitor areas, public events, private properties and borders, and consequently take appropriate actions.

- **Storage of Potentially Relevant Activities.** Multimedia sensors could infer and record potentially relevant activities (thefts, car accidents, traffic violations), and make video/audio streams or reports available for future query.

- **Traffic Avoidance, Enforcement and Control Systems.** It will be possible to monitor car traffic in big cities or highways and deploy services that offer traffic routing advice to avoid congestion. In addition, smart parking advice systems based on WMSANs [23] will allow monitoring available parking spaces and provide drivers with automated parking advice, thus improving mobility in urban areas. Moreover, multimedia sensors may monitor the flow of vehicular traffic on highways and retrieve aggregate information such as average speed and number of cars. Sensors could also detect violations and transmit video streams to law enforcement agencies to identify the violator, or buffer images and streams in case of accidents for subsequent accident scene analysis.

- **Advanced Health Care Delivery.** Telemedicine sensor networks [53] can be integrated with 3G multimedia networks to provide ubiquitous health care services. Patients will carry medical sensors to monitor parameters such as body temperature, blood pressure, pulse oximetry, ECG, breathing activity. Furthermore, remote medical centers will perform advanced remote monitoring of their patients via video and

audio sensors, location sensors, motion or activity sensors, which can also be embedded in wrist devices [53].

- **Structural Health Monitoring.** Multimedia streams can be used to monitor the structural health of bridges [18] or other structures.

- **Industrial Process Control.** Multimedia content such as imaging, temperature, or pressure amongst others, may be used for time-critical industrial process control. *Machine vision* is the application of computer vision techniques to industry and manufacturing, where information can be extracted and analyzed by WMSANs to support a manufacturing process such as those used in semiconductor chips, automobiles, food or pharmaceutical products. For example, in quality control of manufacturing processes, details or final products are automatically inspected to find defects. In addition, machine vision systems can detect the position and orientation of parts of the product to be picked up by a robotic arm. The integration of machine vision systems with WMSANs can simplify and add flexibility to systems for visual inspections and automated actions that require high-speed, high-magnification, and continuous operation.

As observed in [29], WMSANs will stretch the horizon of traditional monitoring and surveillance systems by:

- **Enlarging the View.** The Field of View (FoV) of a single fixed camera, or the Field of Regard (FoR) of a single moving pan-tilt-zoom (PTZ) camera are limited. Instead, a distributed system of multiple cameras and sensors enables perception of the environment from multiple disparate viewpoints, and helps overcoming occlusion effects.

- **Enhancing the View.** The redundancy provided by multiple, possibly heterogeneous, overlapped sensors can provide enhanced understanding and monitoring of

the environment. Overlapped cameras can provide different views of the same area or target, while the joint operation of cameras and audio or infrared sensors can help disambiguate cluttered situations.

- **Enabling Multi-resolution Views.** Heterogeneous media streams with different granularity can be acquired from the same point of view to provide a multi-resolution description of the scene and multiple levels of abstraction. For example, static medium-resolution camera views can be enriched by views from a zoom camera that provides a high-resolution view of a region of interest. For example, such feature could be used to recognize people based on their facial characteristics.

Many of the above applications require the sensor network paradigm to be re-thought in view of the need for mechanisms to deliver multimedia content with a certain level of quality of service (QoS). Since the need to minimize the energy consumption has driven most of the research in sensor networks so far, mechanisms to efficiently deliver application-level QoS, and to map these requirements to network-layer metrics such as latency and jitter, have not been primary concerns in mainstream research on classical sensor networks.

Conversely, algorithms, protocols and techniques to deliver multimedia content over large-scale networks have been the focus of intensive research in the last twenty years, especially in ATM wired and wireless networks. Later, many of the results derived for ATM networks have been re-adapted, and architectures such as Diffserv and Intserv for Internet QoS delivery have been developed. However, there are several main peculiarities that make QoS delivery of multimedia content in sensor networks an even more challenging, and largely unexplored, task:

- **Resource Constraints.** Sensor devices are constrained in terms of battery, memory, processing capability, and achievable data rate [15]. Hence, efficient use of these scarce resources is mandatory.

- **Variable Channel Capacity.** While in wired networks the capacity of each link is

assumed to be fixed and predetermined, in multi-hop wireless networks, the attainable capacity of each wireless link depends on the interference level perceived at the receiver. This, in turn, depends on the interaction of several functionalities that are distributively handled by all network devices such as power control, routing, and rate policies. Hence, capacity and delay attainable at each link are location dependent, vary continuously, and may be bursty in nature, thus making QoS provisioning a challenging task.

- **Cross-layer Coupling of Functionalities.** In multi-hop wireless networks, there is a strict interdependence among functions handled at all layers of the communication stack. Functionalities handled at different layers are inherently and strictly coupled due to the shared nature of the wireless communication channel. Hence, the various functionalities aimed at QoS provisioning should not be treated separately when efficient solutions are sought.

- **Multimedia In-network Processing.** Processing of multimedia content has mostly been approached as a problem isolated from the network-design problem, with a few exceptions such as joint source-channel coding [35] and channel-adaptive streaming [47]. Hence, research that addressed the content delivery aspects has typically not considered the characteristics of the source content and has primarily studied cross-layer interactions among lower layers of the protocol stack. However, the processing and delivery of multimedia content are not independent and their interaction has a major impact on the levels of QoS that can be delivered. WMSANs will allow performing multimedia in-network processing algorithms on the raw data. Hence, the QoS required at the application level will be delivered by means of a combination of both cross-layer optimization of the communication process, and in-network processing of raw data streams that describe the phenomenon of interest from multiple

views, with different media, and on multiple resolutions. Hence, it is necessary to develop application-independent and self-organizing architectures to flexibly perform in-network processing of multimedia contents.

Efforts from several research areas will need to converge to develop efficient and flexible WMSANs, and this in turn, will significantly enhance our ability to interact with the physical environment. These include advances in the understanding of energy-constrained wireless communications, and the integration of advanced multimedia processing techniques in the communication process, and integration with other disciplines such as automation and robotics. Another crucial issue is the development of flexible system architectures and software to allow querying the network to specify the required service (thus providing abstraction from implementation details). At the same time, it is necessary to provide the service in the most efficient way, which may be in contrast with the need for abstraction.

In this chapter, before proposing our solution for WMSANs, we discuss the state of the art in algorithms, protocols, and hardware for the development of wireless multimedia sensor and actor networks, and outline future research challenges in detail. In particular, in Section 4.2 we point out the characteristics of wireless multimedia sensor and actor networks, i.e., the major factors influencing their design. In Section 4.3, we suggest possible architectures for WMSANs and describe their salient features. In Section 4.4, we discuss and classify existing hardware and prototypal implementations for WMSANs, while in Section 4.5 we discuss possible advantages and challenges of multimedia in-network processing.

4.2 Factors influencing the design of Multimedia Sensor and Actor Networks

Wireless Multimedia Sensor and Actor Networks (WMSANs) will be enabled by the convergence of communication and computation with signal processing and several branches

of control theory and embedded computing. This cross-disciplinary research will enable distributed systems of heterogeneous embedded devices that sense, interact, and control the physical environment. There are several factors that mainly influence the design of a WMSAN, which are outlined in this section.

- **Application-specific QoS Requirements.** The wide variety of applications envisaged on WMSANs will have different requirements. In addition to data delivery modes typical of scalar sensor networks, multimedia data include *snapshot* and *streaming multimedia* content. Snapshot-type multimedia data contain event triggered observations obtained in a short time period. Streaming multimedia content is generated over longer time periods and requires sustained information delivery. Hence, a strong foundation is needed in terms of hardware and supporting high-level algorithms to deliver QoS and consider application-specific requirements. These requirements may pertain to multiple domains and can be expressed, amongst others, in terms of a combination of bounds on energy consumption, delay, reliability, distortion, or network lifetime.

- **Convergence of Sensing and Actuation.** The challenges brought about by WMSANs are not to be limited to resource allocation problems. In fact, WMSANs enable new application scenarios in synergy with other research areas. For example, *Distributed Robotics* [24] has been a hot research topic since the mid 90's. In distributed robotics, a task is not completed by a single robot but by a *team of collaborating robots*. Information about the surrounding environment is usually gathered by *onboard sensors*, and team members exchange sensor information to move or perform actions (e.g., collaborate to manipulate heavy objects). As opposed to a single robot, a team of robots can perceive the environment from *multiple disparate viewpoints*. In WMSANs, the ability of the actors to perceive the environment can be pushed one step further: a dense spatio-temporal sampling of the environment, provided by

a pre-deployed sensor network, can be exploited by the whole team of actors, thus increasing the ability of the team to accurately interact with the physical environment. Furthermore, multimedia content gathered by sensors can be used to provide the team of actors with accurate vision from multiple perspectives, while as of today collaborating actors mostly rely on expensive onboard cameras. Clearly, further research is needed to fully leverage the opportunities offered by the integration of actors and multimedia sensors in a wireless network.

- **High Bandwidth Demand.** Multimedia content, especially video streams, require transmission bandwidth that is orders of magnitude higher than that supported by currently available sensors. For example, the nominal transmission rate of state-of-the-art IEEE 802.15.4 compliant components such as Crossbow's [3] MICAz or TelosB [6] motes is $250\,\mathrm{Kbit/s}$. Data rates at least one order of magnitude higher may be required for high-end multimedia sensors, with comparable power consumption. Hence, high data rate and low power consumption transmission techniques need to be leveraged. In this respect, the Ultra Wide Band (UWB) transmission technique seems particularly promising for WMSANs, and its applicability is discussed in Chapter 5.

- **Multimedia Source Coding Techniques.** Uncompressed raw video streams require excessive bandwidth for a multi-hop wireless environment. For example, a single monochrome frame in the NTSC-based *Quarter Common Intermediate Format* (QCIF, 176x120), requires around $21\,\mathrm{Kbytes}$, and at 30 frames per second (fps), a video stream requires over $5\,\mathrm{Mbit/s}$. Hence, it is apparent that efficient processing techniques for lossy compression are necessary for multimedia sensor networks. Traditional video coding techniques used for wireline and wireless communications are based on the idea of reducing the bit rate generated by the source encoder by exploiting source statistics. To this aim, encoders rely on *intra-frame* compression

techniques to reduce redundancy within one frame, while they leverage *inter-frame* compression (also known as *predictive encoding* or *motion estimation*) to exploit redundancy among subsequent frames to reduce the amount of data to be transmitted and stored, thus achieving good rate-distortion performance. Since predictive encoding requires complex encoders, powerful processing algorithms, and entails high energy consumption, it may not be suited for low-cost multimedia sensors. However, it has recently been shown [46] that the traditional balance of complex encoder and simple decoder can be reversed within the framework of the so-called *distributed source coding*, which exploits the source statistics at the decoder, and by shifting the complexity at this end, allows the use of simple encoders. Clearly, such algorithms are very promising for WMSANs and especially for networks of video sensors, where it may not be feasible to use existing video encoders at the source node due to processing and energy constraints.

- **Multimedia In-network Processing.** WMSANs allow performing multimedia in-network processing algorithms on the raw data extracted from the environment. This requires new architectures for collaborative, distributed, and resource-constrained processing that allow for filtering and extraction of semantically relevant information at the edge of the sensor network. This may increase the system scalability by reducing the transmission of redundant information, merging data originated from multiple views, on different media, and with multiple resolutions. For example, in video security applications, information from uninteresting scenes can be compressed to a simple scalar value or not be transmitted altogether, while in environmental applications, distributed filtering techniques can create a time-elapsed image [104]. Hence, it is necessary to develop application-independent architectures to flexibly perform in-network processing of the multimedia content gathered from the environment. For example, IrisNet [81] uses application-specific filtering of sensor feeds at the source, i.e., each application processes its desired sensor feeds on the CPU of the sensor

nodes where data are gathered. This dramatically reduces the bandwidth consumed, since instead of transferring raw data, IrisNet sends only a potentially small amount of processed data. However, the cost of multimedia processing algorithms may be prohibitive for low-end multimedia sensors. Hence, it is necessary to develop scalable and energy-efficient distributed filtering architectures to enable processing of redundant data as close as possible to the periphery of the network.

- **Power Consumption.** Power consumption is a fundamental concern in WMSANs, even more than in traditional wireless sensor networks. In fact, sensors are battery-constrained devices, while multimedia applications produce high volumes of data, which require high transmission rates, and extensive processing. While the energy consumption of traditional sensor nodes is known to be dominated by the communication functionalities, this may not necessarily be true in WMSANs. Therefore, protocols, algorithms and architectures to maximize the network lifetime while providing the QoS required by the application are a critical issue.

- **Flexible Architecture to Support Heterogeneous Applications.** WMSAN architectures will support several heterogeneous and independent applications with different requirements. It is necessary to develop flexible, hierarchical architectures that can accommodate the requirements of all these applications in the same infrastructure.

- **Multimedia Coverage.** Some multimedia sensors, in particular video sensors, have larger sensing radii and are sensitive to direction of acquisition (directivity). Furthermore, video sensors can capture images only when there is unobstructed line of sight between the event and the sensor. Hence, coverage models developed for traditional wireless sensor networks are not sufficient for pre-deployment planning of a multimedia sensor network.

- **Integration with Internet (IP) Architecture.** It is of fundamental importance for the commercial development of sensor networks to provide services that allow querying the network to retrieve useful information from anywhere and at any time. For this reason, future WMSANs will be remotely accessible from the Internet, and will therefore need to be integrated with the IP architecture. The characteristics of WSNs rule out the possibility of all-IP sensor networks and recommend the use of application-level gateways or overlay IP networks as the best approach for integration between WSNs and the Internet [113].

- **Integration with Other Wireless Technologies.** Large scale sensor networks may be created by interconnecting local "islands" of sensors through other wireless technologies. This needs to be achieved without sacrificing on the efficiency of the operation within each individual technology.

4.3 Network Architecture

The problem of designing a *scalable network architecture* is of primary importance. Most proposals for wireless sensor networks are based on a flat, homogenous architecture in which every sensor has the same physical capabilities and can only interact with neighboring sensors. Traditionally, the research on algorithms and protocols for sensor networks has focused on *scalability*, i.e., how to design solutions whose applicability would not be limited by the growing size of the network. Flat topologies may not always be suited to handle the amount of traffic generated by multimedia applications including audio and video. Likewise, the processing power required for data processing and communications, and the power required to operate it, may not be available on each node.

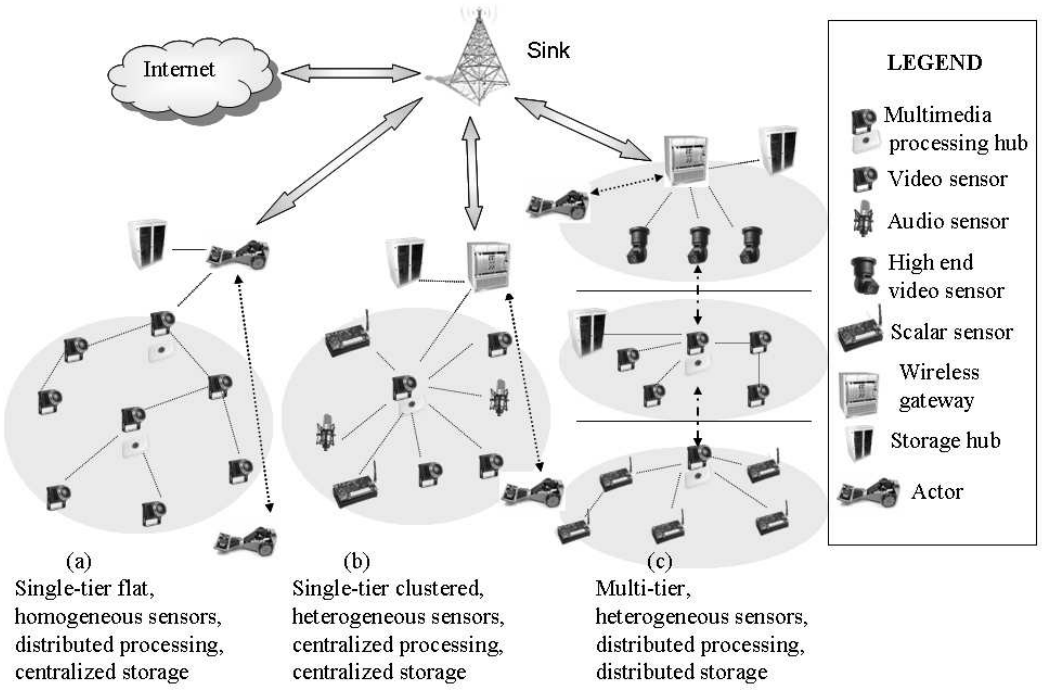


Figure 35: Reference architecture of a Wireless Multimedia Sensor and Actor Network.

4.3.1 Reference Architecture

In Fig. 35, we introduce a reference architecture for WMSANs, where three sensor networks with different characteristics are shown, possibly deployed in different physical locations. The first cloud on the left shows a single-tier network of homogeneous video sensors. A subset of the deployed sensors have higher processing capabilities, and are thus referred to as *processing hubs*. The union of the processing hubs constitutes a distributed processing architecture. The multimedia content gathered is relayed to a *wireless gateway* through a multi-hop path. The gateway is interconnected to a *storage hub*, that is in charge of storing multimedia content locally for subsequent retrieval. Clearly, more complex architectures for distributed storage can be implemented when allowed by the environment and the application needs, which may result in energy savings since by storing it locally, the multimedia content does not need to be wirelessly relayed to remote locations. The wireless gateway is also connected to a central *sink*, which implements the software front-end for network querying and tasking, and to the actor to reactively act on the environment.

The second cloud represents a single-tiered clustered architecture of heterogeneous sensors (only one cluster is depicted). Video, audio, and scalar sensors relay data to a central clusterhead, which is also in charge of performing intensive multimedia processing on the data (processing hub). The clusterhead relays the gathered content to the wireless gateway and to the storage hub. The last cloud on the right represents a multi-tiered network, with heterogeneous sensors. Each tier is in charge of a subset of the functionalities. Resource-constrained, low-power scalar sensors are in charge of performing simpler tasks, such as detecting scalar physical measurements, while resource-rich, high-power devices are responsible for more complex tasks. Data processing and storage can be performed in a distributed fashion at each different tier.

4.3.2 Single-tier vs. Multi-tier Sensor Deployment

One possible approach for designing a multimedia sensor application is to deploy homogeneous sensors and program each sensor to perform all possible application tasks. Such an approach yields a flat, single-tier network of homogeneous sensor nodes. An alternative, multi-tier approach is to use heterogeneous elements [63]. In this approach, resource-constrained, low-power elements are in charge of performing simpler tasks, such as detecting scalar physical measurements, while resource-rich, high-power devices take on more complex tasks. For instance, a surveillance application can rely on low-fidelity cameras or scalar acoustic sensors to perform motion or intrusion detection, while high-fidelity cameras can be woken up on-demand for object recognition and tracking. In [62], a multi-tier architecture is advocated for video sensor networks for surveillance applications. The architecture is based on multiple tiers of cameras with different functionalities, with the lower tier constituted of low-resolution imaging sensors, and the higher tier composed of high-end pan-tilt-zoom cameras. It is argued, and shown by means of experiments, that such an architecture offers considerable advantages with respect to a single-tier architecture in terms of scalability, lower cost, better coverage, higher functionality, and better reliability.

4.3.3 Coverage

In traditional WSNs, sensor nodes collect information from the environment within a predefined *sensing range*, i.e., a roughly circular area defined by the type of sensor being used.

Multimedia sensors generally have larger sensing radii and are also sensitive to the direction of data acquisition. In particular, cameras can capture images of objects or parts of regions that are not necessarily close to the camera itself. However, the image can obviously be captured only when there is an unobstructed line-of-sight between the event and the sensor. Furthermore, each multimedia sensor/camera perceives the environment or the observed object from a different and unique viewpoint, given the different orientations and positions of the cameras relative to the observed event or region. In [102], a preliminary investigation of the coverage problem for video sensor networks is conducted. The concept of sensing range is replaced with the camera's *field of view*, i.e., the maximum volume visible from the camera. It is also shown how an algorithm designed for traditional sensor networks does not perform well with video sensors in terms of coverage preservation of the monitored area.

4.4 Multimedia Sensor and Actor Hardware

In this section, we review and classify existing imaging, multimedia, and processing wireless devices that will find application in next-generation wireless multimedia sensor networks. In particular, we discuss existing hardware, with a particular emphasis on video capturing devices, review existing implementations of multimedia sensor networks, and discuss current possibilities for *energy harvesting* for multimedia sensor devices.

4.4.1 Enabling Hardware Platforms

High-end pan-tilt-zoom cameras and high resolution digital cameras are widely available on the market. However, while such sophisticated devices can find application as high-quality tiers of multimedia sensor networks, we concentrate on low-cost, low-energy consumption imaging and processing devices that will be densely deployed and provide detailed visual information from multiple disparate viewpoints, help overcoming occlusion effects, and thus enable enhanced interaction with the environment.

4.4.1.1 Low-resolution Imaging Motes

The recent availability of CMOS imaging sensors [56] that capture and process an optical image within a single integrated chip, thus eliminating the need for many separate chips required by the traditional charged-coupled device (CCD) technology, has enabled the massive deployment of low-cost visual sensors. CMOS image sensors are already in many industrial and consumer sectors, such as cell phones, personal digital assistants (PDAs), consumer and industrial digital cameras. CMOS image quality is now matching CCD quality in the low- and mid-range, while CCD is still the technology of choice for high-end image sensors. The CMOS technology allows integrating a lens, an image sensor and image processing algorithms, including image stabilization and image compression, on the same chip. With respect to CCD, cameras are smaller, lighter, and consume less power. Hence, they constitute a suitable technology to realize imaging sensors to be interfaced with wireless motes.

However, existing CMOS imagers are still designed to be interfaced with computationally rich host devices, such as cell phones or PDAs. For this reason, the objective of the Cyclops module [88] is to fill the gap between CMOS cameras and computationally constrained devices. Cyclops is an electronic interface between a CMOS camera module and a wireless mote such as MICA2 or MICAz, and contains programmable logic and

memory for high-speed data communication. Cyclops consists of an imager (CMOS Agilent ADCM-1700 CIF camera), an 8-bit ATMEL ATmega128L micro controller (MCU), a complex programmable logic device (CPLD), an external SRAM and an external Flash. The MCU controls the imager, configures its parameters, and performs local processing on the image to produce an inference. Since image capture requires faster data transfer and address generation than the $4\,\mathrm{MHz}$ MCU used, a CPLD is used to provide access to the high speed clock. Cyclops firmware is written in the nesC language [42], based on the TinyOS libraries. The module is interfaced to a host mote to which it provides a high level interface that hides the complexity of the imaging device to the host mote. Moreover, it can perform simple inference on the image data and present it to the host.

Researchers at Carnegie Mellon University are developing the CMUCAM 3, which is an embedded camera endowed with a CIF Resolution (352x288) RGB color sensor that can load images into memory at 26 frames per second. CMUCAM 3 has software JPEG compression and has a basic image manipulation library, and can be interface with an 802.15.4 compliant TelosB mote [6].

In [33], the design of an integrated mote for wireless image sensor networks is described. The design is driven by the need to endow motes with adequate processing power and memory size for image sensing applications. It is argued that 32-bit processors are better suited for image processing than their 8-bit counterpart, which is used in most existing motes. It is shown that the time needed to perform operations such as 2-D convolution on an 8-bit processor such as the ATMEL ATmega128 clocked at $4\,\mathrm{MHz}$ is 16 times higher than with a 32-bit ARM7 device clocked at $48\,\mathrm{MHz}$, while the power consumption of the 32-bit processor is only 6 times higher. Hence, an 8-bit processor turns out to be slower and more energy-consuming. Based on these premises, a new image mote is developed based on an ARM7 32-bit CPU clocked at $48\,\mathrm{MHz}$, with external FRAM or Flash memory, 802.15.4 compliant Chipcon CC2420 radio, that is interfaced with mid-resolution ADCM-1670 CIF CMOS sensors and low-resolution 30x30 pixel optical sensors.

The same conclusion is drawn in [71], where the energy consumption of the 8-bit Atmel AVR processor clocked at $8\,\mathrm{MHz}$ is compared to that of the PXA255 32-bit Intel processor, embedded on a Stargate platform [9] and clocked at $400\,\mathrm{MHz}$. Three representative algorithms are selected as benchmarks, i.e., the cyclic redundancy check, a finite impulse response filter, and a fast Fourier transform. Surprisingly, it is shown that even for such relatively simple algorithms the energy consumption of an 8-bit processor is between one and two orders of magnitude higher.

4.4.1.2 Medium-resolution Imaging Motes Based on the Stargate Platform

Intel has developed several prototypes that constitute important building platform for WMSAN applications. The Stargate board [9] is a high-performance processing platform designed for sensor, signal processing, control, robotics, and sensor network applications. It is designed by Intel and produced by Crossbow. Stargate is based on Intel's PXA-255 Xscale $400\,\mathrm{MHz}$ RISC processor, which is the same processor found in many handheld computers including the Compaq IPAQ and the Dell Axim. Stargate has $32\,\mathrm{Mbyte}$ of Flash memory, $64\,\mathrm{MByte}$ of SDRAM, and an on-board connector for Crossbow's MICA2 or MICAz motes as well as PCMCIA Bluetooth or IEEE 802.11 cards. Hence, it can work as a wireless gateway and as a computational hub for in-network processing algorithms. When connected with a webcam or other capturing device, it can function as a medium-resolution multimedia sensor, although its energy consumption is still high, as documented in [68]. Moreover, although efficient software implementations exist, Xscale processors do not have hardware support for floating point operations, which may be needed to efficiently perform multimedia processing algorithms.

Intel has also developed two prototypal generations of wireless sensors, known as Imote and Imote2. Imote is built around an integrated wireless microcontroller consisting of an 8-bit $12\,\mathrm{MHz}$ ARM7 processor, a Bluetooth radio, $64\,\mathrm{KByte}$ RAM and $32\,\mathrm{Kbyte}$ FLASH memory, as well as several I/O options. The software architecture is based on an ARM port

of TinyOS. The second generation of Intel motes has a common core to the next generation Stargate 2 platform, and is built around a new low-power 32-bit PXA271 XScale processor at $320/416/520\,\mathrm{MHz}$, which enables performing DSP operations for storage or compression, and an IEEE 802.15.4 ChipCon CC2420 radio. It has large on-board RAM and Flash memories ($32\,\mathrm{MByte}$), additional support for alternate radios, and a variety of high speed I/O to connect digital sensors or cameras. Its size is also very limited, $48\mathrm{x}33\,\mathrm{mm}$, and it can run the Linux operating system and Java applications.

4.4.2 Energy Harvesting

As mentioned before, techniques for prolonging the lifetime of battery-powered sensors have been the focus of a vast amount of literature in sensor networks. These techniques include hardware optimizations such as dynamic optimization of voltage and clock rate, wake-up procedures to keep electronics inactive most of the time, and energy-aware protocol development for sensor communications. In addition, *energy-harvesting* techniques, which extract energy from the environment where the sensor itself lies, offer another important mean to prolong the lifetime of sensor devices.

Systems able to perpetually power sensors based on simple COTS photovoltaic cells coupled with supercapacitors and rechargeable batteries have been already demonstrated [59]. In [83], the state of the art in more unconventional techniques for energy harvesting (also referred to as *energy scavenging*) is surveyed. Technologies to generate energy from background radio signals, thermoelectric conversion, vibrational excitation, and the human body, are overviewed.

As far as collecting energy from background radio signals is concerned, unfortunately, an electric field of $1\,\mathrm{V/m}$ yields only $0.26\,\mu\mathrm{W/cm^2}$, as opposed to $100\,\mu\mathrm{W/cm^2}$ produced by a crystalline silicon solar cell exposed to bright sunlight. Electric fields of intensity of a few volts per meter are only encountered close to strong transmitters. Another practice, which consists in broadcasting RF energy deliberately to power electronic devices, is

severely limited by legal limits set by health and safety concerns.

While thermoelectric conversion may not be suitable for wireless devices, harvesting energy from vibrations in the surrounding environment may provide another useful source of energy. Vibrational magnetic power generators based on moving magnets or coils may yield powers that range from tens of microwatts when based on micro-electro mechanical system (MEMS) technologies to over a milliwatt for larger devices. Other vibrational microgenerators are based on charged capacitors with moving plates, and depending on their excitation and power conditioning, yield power on the order of $10\,\mu\mathrm{W}$. In [83], it is also reported that recent analysis [78] suggested that $1\,\mathrm{cm}^3$ vibrational microgenerators can be expected to yield up to $800\,\mu\mathrm{W}/\mathrm{cm}^3$ from machine-induced stimuli, which is orders of magnitude higher than what provided by currently available microgenerators. Hence, this is a promising area of research for small battery-powered devices.

While these techniques may provide an additional source of energy and help prolong the lifetime of sensor devices, they yield power that is several orders of magnitude lower as compared to the power consumption of state-of-the-art multimedia devices. Hence, they may currently be suitable only for very-low duty cycle devices.

4.4.3 Examples of Deployed Multimedia Sensor Networks

There have been several recent experimental studies, mostly limited to video sensor networks. Panoptes [38] is a system developed for environmental observation and surveillance applications, based on Intel StrongARM PDA platforms with a Logitech webcam as a video capture device. Here, video sensors are high-end devices with Linux operating system, $64\,\mathrm{Mbytes}$ of memory, and are networked through 802.11 networking cards. The system includes spatial compression (but not temporal), distributed filtering, buffering, and adaptive priorities for the video stream.

In [28], a system whose objective is to limit the computation, bandwidth, and human attention burdens imposed by large-scale video surveillance systems is described. In-network

processing is used on each camera to filter out uninteresting events locally, avoiding disambiguation and tracking of irrelevant environmental distractors. A resource allocation algorithm is also proposed to steer pan-tilt cameras to follow interesting targets while maintaining awareness of possibly emerging new targets.

In [63], the design and implementation of SensEye, a multi-tier network of heterogeneous wireless nodes and cameras, is described. The surveillance application consists of three tasks: object detection, recognition and tracking. The objective of the design is to demonstrate that a camera sensor network containing heterogeneous elements provides numerous benefits over traditional homogeneous sensor networks. For this reason, SensEye follows a three-tier architecture, as shown in Fig. 36. The lowest tier consists of low-end devices, i.e., MICA2 Motes equipped with $900\,\mathrm{MHz}$ radios interfaced with scalar sensors, e.g., vibration sensors. The second tier is made up of motes equipped with low-fidelity Cyclops [88] or CMUcam [94] camera sensors. The third tier consists of Stargate [9] nodes equipped with webcams. Each Stargate is equipped with an embedded $400\,\mathrm{MHz}$ XScale processor that runs Linux and a webcam that can capture higher fidelity images than tier 2 cameras. Tier 3 nodes also perform gateway functions, as they are endowed with a low data rate radio to communicate with motes in tiers 1-2 at $900\,\mathrm{MHz}$, and an 802.11 radio to communicate with tier 3 Stargate nodes. An additional fourth tier may consist of a sparse deployment of high-resolution, high-end pan-tilt-zoom cameras connected to embedded PCs. The camera sensors at this tier can be used to track moving objects, and can be utilized to fill coverage gaps and provide additional redundancy. The underlying design principle is to map each task requested by the application to the lowest tier with sufficient resources to perform the task. Devices from higher tiers are woken up on-demand only when necessary. For example, a high-resolution camera can be woken up to retrieve high resolution images of an object that has been previously detected by a lower tier. It is shown that the system can achieve an order of magnitude reduction in energy consumption while providing comparable surveillance accuracy with respect to single-tier surveillance

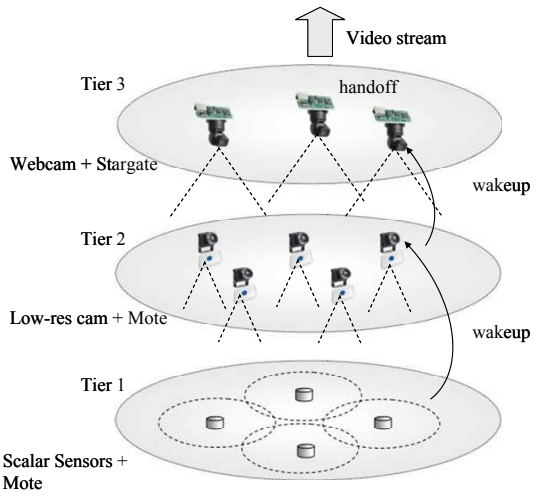


Figure 36: The Multi-tier architecture of Senseye [63].

systems.

In [68], experimental results on the energy consumption of a video sensor network testbed are presented. Each sensing node in the testbed consists of a Stargate board equipped with an 802.11 wireless network card and a Logitech QuickCam Pro 4000 webcam. The energy consumption is assessed using a benchmark that runs basic tasks such as processing, flash memory access, image acquisition, and communication over the network. Both steady-state and transient energy consumption behavior obtained by direct measurements of current with a digital multimeter are reported. In the steady state, it is shown that communication-related tasks are less energy-consuming than intensive processing and flash access when the radio modules are loaded. Interestingly, and unlike in traditional wireless sensor networks [85], the processing-intensive benchmark results in the highest current requirement, and transmission is shown to be only about 5% more energy-consuming than reception. Experimental results also show that delay and additional amount of energy consumed due to transitions (e.g., to go to sleep mode) are not negligible and must be accounted for in network and protocol design.

IrisNet (Internet-scale Resource-Intensive Sensor Network Services) [81] is an example software platform to deploy heterogeneous services on WMSANs. IrisNet allows harnessing a global, wide-area sensor network by performing Internet-like queries on this infrastructure. Video sensors and scalar sensors are spread throughout the environment, and collect potentially useful data. IrisNet allows users to perform Internet-like queries to video sensors and other data. The user views the sensor network as a single unit that can be queried through a high-level language. Each query operates over data collected from the global sensor network, and allows simple Google-like queries as well as more complex queries involving arithmetic and database operators.

The architecture of IrisNet is two-tiered: heterogeneous sensors implement a common shared interface and are called *sensing agents* (SA), while the data produced by sensors is stored in a distributed database that is implemented on *organizing agents* (OA). Different *sensing services* are run simultaneously on the architecture. Hence, the same hardware infrastructure can provide different sensing services. For example, a set of video sensors can provide a parking space finder service, as well as a surveillance service. Sensor data is represented in the Extensible Markup Language (XML), which allows easy organization of hierarchical data. A group of OAs is responsible for a sensing service, collects data produced by that service, and organizes the information in a distributed database to answer the class of relevant queries. Irisnet also allows programming sensors with filtering code that processes sensor readings in a service-specific way. A single SA can execute several such software filters (called senselets) that process the raw sensor data based on the requirements of the service that needs to access the data. After senselet processing, the distilled information is sent to a nearby OA.

4.4.4 Broadband and Wireless Networking Lab WMSAN Testbed

We have recently built an experimental testbed at the Broadband and Wireless Networking (BWN) Laboratory at Georgia Tech based on currently-off-the-shelf advanced devices to

Figure 37: Stargate board interfaced with a medium resolution camera. Stargate hosts an 802.11 card and a MICAz mote that functions as a gateway to the sensor network.

demonstrate the efficiency of algorithms and protocols for multimedia communications through wireless sensor and actor networks.

The testbed is integrated with our scalar sensor network testbed, which is composed of a heterogeneous collection of imotes from Intel and MICAz motes from Crossbow. Although our testbed already includes 60 scalar sensors, we plan to increase its size to deploy a higher-scale testbed that allows testing more complex algorithms and assess the scalability of the communication protocols under examination.

The WMSAN-testbed includes three different types of multimedia sensors: low-end imaging sensors, medium-quality webcam-based multimedia sensors, and pan-tilt cameras mounted on mobile robots.

Low-end imaging sensors such as CMOS cameras can be interfaced with Crossbow MICAz motes. Medium-end video sensors are based on Logitech webcams interfaced with Stargate platforms (see Fig. 37).

The high-end video sensors consist of pan-tilt cameras installed on an Acroname GARCIA robotic platform [1] (actor), and shown in Fig. 38. Actors constitute a mobile platform that can perform adaptive sampling based on event features detected by low-end motes. The mobile actor can redirect high-resolution cameras to a region of interest when events are

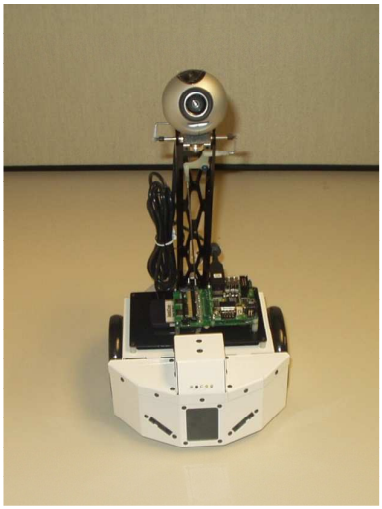


Figure 38: Acroname GARCIA, a mobile robot with a mounted pan-tilt camera and endowed with 802.11 as well as ZigBee interfaces.

Figure 39: GARCIA deployed on the sensor test-bed. It acts as a mobile sink, and can move to the area of interest for closer visual inspection. It can also coordinate with other actors and has built-in collision avoidance capability.

detected by lower-tier, low-resolution video sensors that are densely deployed, as seen in Fig. 39.

The testbed also includes storage and computational hubs, which are needed to store large multimedia content and perform computationally intensive multimedia processing algorithms.

4.5 Collaborative In-network Processing

As discussed previously, collaborative in-network multimedia processing techniques are of great interest in the context of a WMSAN. It is necessary to develop architectures and algorithms to flexibly perform these functionalities *in-network* with minimum energy consumption and limited execution time. The objective is usually to avoid transmitting large amounts of raw streams to the sink by processing the data in the network to reduce the communication volume.

Given a source of data (e.g., a video stream), different applications may require diverse information (e.g., raw video stream vs. simple scalar or binary information inferred by processing the video stream). This is referred to as *application-specific querying and processing*. Hence, it is necessary to develop expressive and efficient querying languages, and to develop distributed filtering and in-network processing architectures, to allow real-time retrieval of useful information.

Similarly, it is necessary to develop architectures that efficiently allow performing data fusion or other complex processing operations *in-network*. Algorithms for both inter-media and intra-media data aggregation and fusion need to be developed, as simple distributed processing schemes developed for existing scalar sensors are not suitable for computation-intensive processing required by multimedia contents. Multimedia sensor networks may require computation-intensive processing algorithms (e.g., to detect the presence of suspicious activity from a video stream). This may require considerable processing to extract

meaningful information and/or to perform compression. A fundamental question to be answered is whether this processing can be done on sensor nodes (i.e., a flat architecture of multifunctional sensors that can perform any task), or if the need for specialized devices, e.g. *computation hubs*, arises.

In what follows, we discuss a non-exhaustive set of significative examples of processing techniques that would be applicable distributively in a WMSAN, and that will likely drive research on architectures and algorithms for distributed processing of raw sensor data.

4.5.1 Data Alignment and Image Registration

Data alignment consists of merging information from multiple sources. One of the most widespread data alignment concepts, image registration [112], is a family of techniques, widely used in areas such as remote sensing, medical imaging, and computer vision, to geometrically align different images (reference and sensed images) of the same scene taken at different times, from different viewpoints, and/or by different sensors:

- *Different Viewpoints (Multiview Analysis).* Images of the same scene are acquired from different viewpoints, to gain a larger 2D view or a 3D representation of the scene of interest. Main applications are in remote sensing, computer vision and 3D shape recovery.

- *Different Times (Multitemporal Analysis).* Images of the same scene are acquired at different times. The aim is to find and evaluate changes in time in the scene of interest. The main applications are in computer vision, security monitoring, and motion tracking.

- *Different Sensors (Multimodal Analysis).* Images of the same scene are acquired by different sensors. The objective is to integrate the information obtained from different source streams to gain more complex and detailed scene representation.

Registration methods usually consist of four steps, i.e., *feature detection*, *feature matching*, *transform model estimation*, and *image re-sampling and transformation*. In feature detection, distinctive objects such as closed-boundary regions, edges, contours, line intersections, corners, etc. are detected. In feature matching, the correspondence between the features detected in the sensed image and those detected in the reference image is established. In transform model estimation, the type and parameters of the so-called mapping functions, which align the sensed image with the reference image, are estimated. The parameters of the mapping functions are computed by means of the established feature correspondence. In the last step, image re-sampling and transformation, the sensed image is transformed by means of the mapping functions.

These functionalities can clearly be prohibitive for a single sensor. Hence, research is needed on how to perform these functionalities on parallel architectures of sensors to produce single data sets.

4.5.2 WMSANs as Distributed Computer Vision Systems

Computer vision is a subfield of artificial intelligence, whose purpose is to allow a computer to extract features from a scene, an image or multidimensional data in general. The objective is to present this information to a human operator or to control some process (e.g., a mobile robot or an autonomous vehicle). The image data that is fed into a computer vision system is often a digital image, a video sequence, a 3D volume from a tomography device or other multimedia content. Traditional computer vision algorithms require extensive computation, which in turn entails high power consumption.

WMSANs enable a new approach to computer vision, where visual observations across the network can be performed by means of distributed computations on multiple, possibly low-end, vision nodes. This requires tools to interface with the user such as new querying languages and abstractions to express complex tasks, that are then distributively accomplished through low-level operations on multiple vision nodes. To this aim, it is necessary

to coordinate computations across the vision nodes and return the integrated results, which will consist of metadata information, to the final user.

In [89], the proposed Deep Vision network performs operations including object detection or classification, image segmentation, and motion analysis through a network of low-end MICA motes equipped with Cyclops cameras [88]. Information such as the presence of an intruder, the number of visitors in a scene or the probability of presence of a human in the monitored area is obtained by collecting the results of these operations. Deep Vision provides a querying interface to the user in the form of declarative queries. Each operation is represented as an attribute that can be executed through an appropriate query. In this way, low-level operations and processing are encapsulated in a high-level querying interface that enables simple interaction with the video network. As an example, the vision network can be deployed in areas with public and restricted access spaces. The task of detecting objects in the restricted-access area can be expressed as a query that requests the result of object detection computations such as:

SELECT *Object,Location*
REPORT=30
FROM *Network*
WHERE *Access = Restricted*
PERIOD=30;

The above query triggers the execution of the object detection process on the vision nodes that are located in the restricted-access areas in $30\,\mathrm{s}$ intervals.

CHAPTER V

CROSS-LAYER QUALITY OF SERVICE SUPPORT FOR ULTRA WIDE BAND WIRELESS MULTIMEDIA SENSOR AND ACTOR NETWORKS

5.1 Preliminaries

As previously discussed, Wireless multimedia sensor networks will not only enhance existing sensor network applications such as tracking, home automation, and environmental monitoring, but will also enable several new applications such as multimedia surveillance sensor networks, traffic congestion avoidance systems, traffic enforcement and control systems, storage of potentially relevant activities, advanced health care delivery, structural health monitoring, and industrial process control.

Many of these applications require the sensor network paradigm to be re-thought in view of the need for mechanisms to deliver multimedia content with a certain level of quality of service (QoS). Since the need to minimize the energy consumption has driven most of the research in sensor networks so far, mechanisms to efficiently deliver application-level QoS, and to map these requirements to network-layer metrics such as latency, have not been primary concerns in mainstream research on sensor networks.

There are several main peculiarities that make QoS delivery of multimedia content in sensor networks a challenging, and largely unexplored, task:

- *Resource Constraints.* Sensor devices are constrained in terms of battery, memory, processing capability, and achievable data rate [15]. At the same time, multimedia applications may have high bandwidth demands. Hence, efficient use of scarce resources is mandatory.

- *Variable Channel Capacity.* The most challenging issue towards QoS provisioning is that, while in wired networks the capacity of each link is assumed to be fixed and predetermined, in multi-hop wireless networks, the attainable capacity of each wireless link depends on the interference level perceived at the receiver. This, in turn depends on the interaction of several functionalities distributively handled by all network devices such as power control, routing, and rate policies. Hence, capacity and delay attainable at each link are location dependent, vary continuously, and may be bursty in nature, thus making QoS provisioning a challenging task.

- *Cross-layer coupling of functionalities.* In multi-hop wireless networks there is a strict interdependence among functions handled at all layers of the communication stack. Functionalities handled at different layers are inherently and strictly coupled due to the wireless nature of the communication channel. Hence, the various functionalities aimed at QoS provisioning should not be treated separately when efficient solutions are sought.

- *Multimedia In-network Processing.* WMSANs allow performing multimedia in-network processing algorithms on the raw data. Hence, the QoS required at the application level can be delivered by means of a combination of both cross-layer optimization of the communication process, and in-network processing of raw data streams that describe the phenomenon of interest from multiple views, with different media, and on multiple resolutions.

The Ultra Wide Band (UWB) [1] technology has the potential to enable low power consumption, high data rate communications within tens of meters, characteristics that make it an ideal choice for WMSANs.

[1]The FCC defines UWB as a signal with either a fractional bandwidth of 20% of the center frequency or 500 MHz (when the center frequency is above 6 GHz). The FCC calculates the fractional bandwidth as $2(f_H - fL)/(f_H + f_L)$ where f_H represents the upper frequency of the -10 dB emission limit and f_L represents the lower frequency limit of the -10 dB emission limit [91].

UWB signals have been used for several decades in the radar community. Recently, the U.S. Federal Communications Commission (FCC) Notice of Inquiry in 1998 and the First Report and Order in 2002 [8] inspired a renewed flourish of research and development efforts in both academy and industry due to the characteristics of UWB that make it a viable candidate for wireless communications in dense multi-path environments.

Although UWB signals, as per the specifications of the FCC, use the spectrum from $3.1\,\mathrm{GHz}$ to $10.6\,\mathrm{GHz}$, with appropriate interference limitation, UWB devices can operate using spectrum occupied by existing radio services without causing interference, thereby permitting scarce spectrum resources to be used more efficiently. Instead of dividing the spectrum into distinct bands that are then allocated to specific services, UWB devices are allowed to operate overlaid and thus interfere with existing services, at a low enough power level that existing services would not experience performance degradation. The First Report and Order by the FCC includes standards designed to ensure that existing and planned radio services, particularly safety services, are adequately protected.

There exist two main variants of UWB. The first, known as Time-Hopping Impulse Radio UWB (TH-IR-UWB) [91], and mainly developed by Win and Scholtz [109], is based on sending very short duration pulses (in the order of hundreds of picoseconds) to convey information. Time is divided into frames, each of which is composed of several chips of very short duration. Each sender transmits one pulse in a chip per frame only, and multi-user access is provided by pseudo-random time hopping sequences (THS) that determine in which chip each user should transmit. A different approach, known as MultiCarrier UWB (MC-UWB), uses multiple simultaneous carriers, and is usually based on Orthogonal Frequency Division Multiplexing (OFDM) [16].

MC-UWB is particularly well-suited for avoiding interference because its carrier frequencies can be precisely chosen to avoid narrowband interference to or from narrowband systems. However, implementing a MC-UWB front-end power amplifier can be challenging due to the continuous variations in power over a very wide bandwidth. Moreover,

when OFDM is used, high-speed FFT processing is necessary, which requires significant processing power and leads to complex transceivers.

TH-IR-UWB signals require fast switching times for the transmitter and receiver and highly precise synchronization. Transient properties become important in the design of the radio and antenna. The high instantaneous power during the brief interval of the pulse helps to overcome interference to UWB systems, but increases the possibility of interference from UWB to narrowband systems. The RF front-end of an TH-IR-UWB system may resemble a digital circuit, thus circumventing many of the problems associated with mixed-signal integrated circuits. Simple TH-IR-UWB systems can be very inexpensive to construct.

Although no sound analytical or experimental comparison between the two technologies is available to our knowledge, we believe that TH-IR-UWB is particularly appealing for WMSANs for the following reasons:

- It enables high data rate, very low power wireless communications, on simple-design, low-cost radios (carrierless, baseband communications) [109];
- Its fine delay resolution properties are appropriate for wireless communications in dense multipath environment, by exploiting more resolvable paths [97];
- Provides large processing gain in presence of interference;
- Provides flexibility, as data rate can be traded for power spectral density and multipath performance;
- Finding suitable codes for THS is trivial (as opposed to CDMA codes), and no assignment protocol is necessary;
- It naturally allows for integrated MAC/PHY solutions; [76]. Moreover, interference mitigation techniques [76] allow realizing MAC protocols that do not require mutual temporal exclusion between different transmitters. In other words, simultaneous

communications of neighboring devices are feasible without complex receivers as required by CDMA;

- The large instantaneous bandwidth enables fine time resolution for accurate position estimation [45] and for network time distribution (synchronization);

- UWB signals have extremely low power spectral density, with low probability of intercept/detection (LPI/D), which is particularly appealing for military covert operations.

Particularly appealing for WMSANs are UWB high data rate with low power consumption, and its positioning capabilities. Positioning capabilities are needed in sensor networks to associate physical meaning to the information gathered by sensors. Moreover, knowledge of the position of each network device allows for scalable routing solutions [74]. While angle-of-arrival techniques and signal strength based techniques do not give particular advantages with respect to other transmission techniques, time-based approaches in UWB allow ranging accuracy in the order of centimeters [45]. This can be intuitively explained by the expression in (60), which gives a lower bound on the best achievable accuracy of a distance estimate $\hat{d}$ [45]:

$$\sqrt{Var(\hat{d})} \geq \frac{c}{2\sqrt{2}\pi\sqrt{SNR}\beta}, \tag{60}$$

where c is the speed of light, SNR represents the signal-to-noise ratio, and β is the effective signal bandwidth. As can be seen, the accuracy of the time-based localization technique can be improved by either increasing the effective bandwidth or the SNR. For this reason, the large bandwidth of UWB systems allows extremely accurate location estimations, e.g., within one inch at $SNR = 0dB$ and with a pulse of $1.5\,\mathrm{GHz}$ bandwidth. Excellent comprehensive surveys of the UWB transmission technique, and of localization techniques for UWB systems, are provided in [110] and [45], respectively.

The IEEE 802.15.3a task group has been discussing for three years an alternate physical layer for its high data rate Wireless Personal Area Networks (WPAN) standard. However, in early 2005 the group has been disbanded after not being able to reach a consensus on a single UWB-based standard between two competing proposal from two leading industry groups, the UWB Forum and the WiMedia Alliance. The UWB Forum proposal was based on a Direct Sequence DS-UWB technology, while the Wimedia alliance was proposing a Multi-band Orthogonal Frequency Division Multiplexing (MB-OFDM). The IEEE 802.15.4a task group is developing an alternate physical layer for low data rate, very low power consumption sensors, based on impulse radio UWB.

While physical layer aspects of UWB communications have received considerable attention in the last few years, higher layer solutions for multi-hop wireless networking with UWB are not mature yet. Although, similarly to CDMA, TH-IR-UWB is a multi-user radio technology, non-zero cross-correlation between time-hopping sequences, time-asynchronicity between sources and the strong effect of multipath propagation require for suitable MAC and higher layer solutions. For the reasons above, in this paper we propose a new cross-layer communication architecture whose objective is to reliably and flexibly *deliver QoS to heterogeneous applications in WMSANs*, by carefully leveraging and controlling interactions among layers according to the applications requirements.

This chapter is organized as follows. In Section 5.2 we discuss principles and previous work on cross-layer design for multi-hop wireless networks. In Section 5.3, we discuss previous work on multi-hop networking with UWB. In Section 5.4, we introduce the considered system model. In Section 5.5 we outline the main design principles, and describe the proposed cross-layer controller. In Section 5.6, we describe the routing and admission control functionalities, while in Section 5.7 we describe the proposed dynamic code assignment and scheduling policies. In Section 5.8 we discuss performance evaluation results.

5.2 Cross-layer Design

As previously discussed, in multi-hop wireless networks there is a strict interdependence among functions handled at all layers of the communication stack. The physical, MAC, and routing layers together impact the contention for network resources. The physical layer has a direct impact on multiple access of nodes in wireless channels by affecting the interference at the receivers. The MAC layer determines the bandwidth allocated to each transmitter, which naturally affects the performance of the physical layer in terms of successfully detecting the desired signals. On the other hand, as a result of transmission schedules, high packet delays and/or low bandwidth can occur, forcing the routing layer to change its route decisions. Different routing decisions alter the set of links to be scheduled, and thereby influence the performance of the MAC layer. Furthermore, congestion control and power control are also inherently coupled [27], as the capacity available on each link depends on the transmission power. Moreover, specifically to multimedia transmissions, the application layer does not require full insulation from lower layers, but needs instead to perform source coding based on information from the lower layers to maximize the multimedia performance. Existing solutions often do not provide adequate support for multimedia applications since the resource management, adaptation, and protection strategies available in the lower layers of the stack are optimized without explicitly considering the specific characteristics of multimedia applications. Similarly, multimedia compression and streaming algorithms do not consider the mechanisms provided by the lower layers for error protection and resource allocation [95].

The additional challenges brought about by real-time streaming of multimedia content in WMSANs call for new research on cross-layer optimization and cross-layer design methodologies, to leverage potential improvements of exchanging information between different layers of the communication stack. However, the increased complexity of functionalities needed to deliver QoS to multimedia applications needs to be managed as well. In particular, it is important to keep some form of logical separation of these functionalities

to preserve upgradability and ease of design and testing. To this aim, it is needed to specify standardized interfaces that will allow leveraging these interactions.

Although a consistent amount of recent papers have focused on cross-layer design and improvement of protocols for WSNs, a systematic methodology to accurately model and leverage cross-layer interactions is still largely missing. In this respect, the design of networking protocols for multi-hop wireless ad hoc and sensor networks can be interpreted as the (possibly distributed) solution of resource allocation problems at different layers. From an engineering perspective, most networking problems can in fact be seen as resource allocation problem, where users (network nodes) are assigned resources (power, time slots, paths, rates, etc.) under some specified system constraints. Resource allocation in the context of multi-hop wireless networks has been extensively studied in the last few years, typically with the objectives of maximizing the network lifetime [25], minimizing the energy consumption [73], or maximizing the network throughput [55]. However, most of the existing studies decompose the resource allocation problem at different layers, and consider allocation of the resources at each layer separately. In most cases, resource allocation problems are treated either heuristically, or without considering cross-layer interdependencies, or by considering pairwise interactions between isolated pairs of layers.

A typical example of the *tight coupling* between functionalities handled at different layers is the interaction between congestion control and power control mechanisms [27]. The congestion control regulates the allowed source rates so that the total traffic load on any link does not exceed the available capacity. In typical congestion control problems, the capacity of each link is assumed to be fixed and predetermined. However, in multi-hop wireless networks, the attainable capacity of each wireless link depends on the interference levels, which in turn depend on the power control policy. Hence, congestion control and power control are inherently coupled and should not be treated separately when efficient solutions are sought.

In [95], the cross-layer transmission of multimedia content over wireless networks is

formalized as an optimization problem. Several different approaches for cross-layer design of multimedia communications are discussed, including *bottom-up approach*, where the lower layers try to insulate the higher layers from losses and channel capacity variations, and *top-down*, where the higher layer protocols optimize their parameters at the next lower layer. However, only single-hop networks are considered.

In [99], several techniques that provide significant performance gains through cross-layer optimizations are surveyed. In particular, the improvements of adaptive link layer techniques such as adaptive modulation and packet size optimization, joint allocation of capacity and flows (i.e., MAC and routing), joint scheduling and rate allocation, are discussed. While still maintaining a strict layered architecture, it is shown how these cross-layer optimizations help improve the spectral efficiency at the physical layer, and the peak signal-to-noise ratio (PSNR) of the video stream perceived by the user. Clearly, energy-constrained multimedia sensors may need to leverage cross-layer interactions one step further. At the same time, optimization metrics in the energy domain need to be considered as well.

5.3 Related Work

There is a vast literature on physical layer aspects of the UWB technology. Excellent comprehensive surveys of the UWB transmission technique, and of localization techniques for UWB systems, are provided in [110] and [45], respectively. As far as multi-hop networking with UWB is concerned, some recent proposals exist, especially at the MAC layer.

In [30], the problem of joint rate and power assignment is formulated as an optimization problem for TH-IR-UWB. It is shown that when the objective is to maximize the aggregate data rate, the optimal solution always corresponds to points where individual devices transmit at the maximum power, or do not transmit at all. The finding is confirmed and justified in [86], where the authors show that power control is not required and may even be suboptimal for wireless networks in the linear regime, i.e., when the achievable data rate

is linearly dependent on the *signal-to-interference-plus-noise ratio* (SINR) at the receiver. Note that this is a peculiar characteristic of TH-IR-UWB, and does not hold in general for narrowband systems such as CDMA or IEEE 802.11. The result holds both for high data rate networks, where the objective is to maximize the data rate under power constraints, and for low-power networks, where the objective is to minimize the power consumption under constraints on minimum data rates.

Based on the above finding, an uncoordinated MAC protocol for low-power UWB devices is proposed in [76]. While most existing protocols manage interference and multiple-access through power control or mutual exclusion, the MAC proposed in [76] is based on rate control, i.e., it dynamically adapts the channel code based on the interference perceived at the receiver. The design takes advantage of the nature of pulsed TH-UWB to further propose an interference mitigation scheme that alleviates the need for an exclusion scheme. Each device is always allowed to transmit and continuously adapts its channel code to the interference experienced at the destination. Such MAC layer does not need coordination among neighbors that are not involved in the communication, and is shown by simulation to achieve a significant increase in network throughput compared to alternative designs. In [58], two MAC packet scheduling schemes are proposed, whose objective is to find a suitable tradeoff between system efficiency and fairness. The rate achieved on each link is approximately proportional to its channel quality level.

The problem of joint optimal power control, scheduling and routing in UWB Networks is dealt with in [87], with the objective of maximizing the aggregate achievable data rate. The problem is formulated as a mathematical optimization problem, and is solved approximately for small topologies (up to 50 nodes). Although the paper does not propose practical solutions, it points out important design principles for UWB networks. In particular, it shows that it is optimal to have an exclusion region around the destination, in which all nodes remain silent during transmission, whereas nodes outside of this region can transmit in parallel, regardless of the interference they produce at the destination. As for the routing,

it shows that relaying along a minimum energy and loss route is always better than using longer hops or sending directly even if the objective is to maximize the data rate. A similar problem is defined in [101], although the problem is formulated for sensor networks instead of general-purpose ad hoc networks. A complex non-linear optimization problem is formulated to assess the feasibility of relaying data from a set of sources to the base station. The emphasis of the paper is mostly on developing an efficient heuristic to solve the problem. The considered physical layer model corresponds to a MC-UWB system based on CDMA. Hence, the achievable rate is not a linear function of the SINR and most results derived in [87] on routing and power control do not hold.

Unlike our work, none of the previously proposed solutions consider the problem of satisfying and differentiating between QoS requirements of the overlying applications. Moreover, no existing practical solution considers the cross-layer interactions between routing, MAC and physical layer functionalities.

5.4 System Model

In this section we describe the system model. In particular, in Section 5.4.1 we describe the considered network model, while in Section 5.4.2 we describe the considered physical layer model in detail. In Section 5.4.3, we describe the multipath channel model adopted. In Section 5.4.4 we introduce notation concerning channel coding of data packets, while in Section 5.4.5 we describe the traffic classes to be supported.

5.4.1 Network Model

The sensor network is represented as a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, ..., v_N\}$ is a finite set of devices (nodes) in a finite-dimension terrain, with $N = |\mathcal{V}|$, and $\mathcal{E}$ is the set of links among nodes, i.e., $e_{ij} \in \mathcal{E}$ iff nodes v_i and v_j are within each other's transmission range. Node v_N (also N for simplicity) represents the destination actor. Each link e_{ij} is also associated with its path loss g_{ij}, which is dependent on the positions $\mathbf{x_i}$ and $\mathbf{x_j}$, respectively, of nodes v_i and v_j (also i and j for simplicity in the following). $\mathcal{S}$ represents the set of

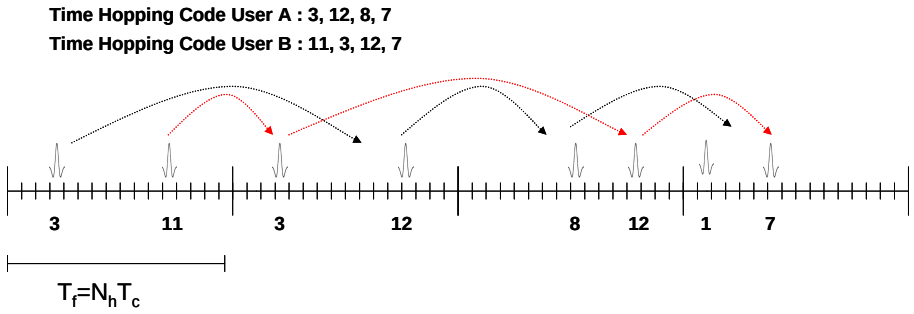


Figure 40: Multiuser access in a TH-IR-UWB system.

sources, which includes those sensors that sense information from the environment to send it to the destination actor N.

5.4.2 Physical Layer Model

TH-IR-UWB transmits subnanosecond pulses (in the order of hundreds of picoseconds), referred to as *monocycles*. We model a monocycle as the second derivative of a Gaussian pulse. Gaussian pulses are generally used as they can easily be implemented in hardware. Time is slotted in chips of duration T_c, and chips are organized in frames of duration $T_f = N_h T_c$, where N_h is the number of chips per frame. Each user transmits one pulse in one chip per frame, and determines in which chip to transmit based on a pseudo-random *time hopping sequence* (THS), as shown in Fig. 40. We assume the train of monocycles to be modulated based on Pulse Position Modulation (PPM), i.e., a '1' symbol is carried by a monocycle delayed by a time δ with respect to the beginning of the chip, while a '0' symbol begins with the chip. In the above model, the signal $s^{(k)}(t,i)$ generated by the k^{th} user to convey the i^{th} symbol can be expressed as

$$s^{(k)}(t,i) = \sqrt{\frac{E_b}{N_0}} \sum_{j=iN_s}^{(i+1)N_s-1} p(t - c_j^{(k)} T_c - jT_f - d_i^{(k)}\delta), \tag{61}$$

where $p(t) = [1 - 4\pi(\frac{t}{\tau_p})^2] \exp(-2\pi(\frac{t}{\tau_p})^2)$ is the second derivative of a Gaussian Pulse, $\{c_j^{(k)}\}$ is the time hopping sequence of the k^{th} source, with $0 \leq c_j^{(k)} \leq N_h - 1$, $\{d_i^{(k)}\}$ is the information-bearing sequence, $d_i^{(k)} \in 0, 1$, E_b represents the energy per bit and N_s the

number of pulses to represent a single bit. Clearly, by increasing the number of pulses per bit N_s one can increase the robustness to multiuser interference, at the expense of the data rate, which can be expressed as $R = 1/N_s T_f$. This technique is also referred to as *pulse repetition coding*. Although simple, pulse repetition codes have coarse granularity, and may thus not be the best choice to develop flexible multi-rate schemes. Each transmitter i transmits at a specified *raw pulse rate* $R_{0,i}$.

Table 4: Ultra Wide Band Physical Layer model

Parameter	**Value**
T_f [s]	$2 \cdot 10^{-7}$
T_c [s]	$0.9 \cdot 10^{-9}$
τ_p [s]	$0.2877 \cdot 10^{-9}$
δ [s]	$0.15 \cdot 10^{-9}$
N_h	221

Theoretical results on the wide-band communication channel show that the optimal signaling consists of sending very short pulses of maximum allowed energy. Hence, assuming that pulses generated at the physical layer have a width T_p, we transmit at a peak power $P_{peak} = E_{peak}/T_p = 0.28\,\mathrm{mW}$, i.e., the limits allowed by regulations and hardware constraints [76].

5.4.3 Multi-path Channel Model

We model the UWB channel according to the model in [79], which was accepted by the IEEE 802.15.4a standardization group for comparison of the proposals submitted to the standardization board. The model, specifically developed for sensor network applications, is based on extensive measurements of UWB channels and can be parameterized for indoor residential, indoor office, outdoor, and industrial environments amongst others. The path loss is expressed as a function of distance and frequency separately, i.e., $g_{ij}(f, d_{ij}) = g(f)g_{ij}(d_{ij})$, with $\sqrt{g_{ij}(f)} \propto f^{-k}$, and with

$$g_{ij}(d_{ij}) = g_0 - 10\alpha \log_{10}(\frac{d_{ij}}{d_0}) \tag{62}$$

where the reference distance d_0 is set to $1\,\mathrm{m}$. The path loss exponent α depends on whether there is line of sight between the transmitter and the receiver or not, on the antenna gain and efficiency. Note that shadowing can be neglected in 802.15.4a simulations.

The impulse response of the channel is modeled according to a modified Saleh-Valenzuela model [79]. The model reproduces the *clustering phenomenon* observed in several UWB measurements, and accordingly assumes that multipath components arrive in clusters, and that there is independent fading for each cluster and for each ray within the cluster. The impulse response of the channel is given by

$$h(t) = \sum_{l=0}^{L} \sum_{m=0}^{M} \alpha_{m,l} u_o(t - T_l - \tau_{m,l}) \tag{63}$$

where $\alpha_{m,l}$ is the multipath gain coefficient for the m^{th} ray in the l^{th} cluster. The interarrival times between two consecutive rays in a cluster and between two consecutive clusters are negative exponentially distributed. Hence, the cluster arrival times follow a distribution $P(T_l|T_{l-1}) = \Lambda \exp(-\Lambda(T_l - T_{l-1})), l > 0$, while ray arrival times follow a Poisson distribution of parameter λ. The number of clusters L is Poisson-distributed. The power delay profile (PDP), i.e., the mean power of the different paths, is exponential within each cluster. Table 4 reports the parameters used for the channel model.

5.4.4 Coding

The proposed system includes a *channel encoder* block that encodes raw data bits into encoded bits that are then transmitted as pulses by the UWB modulator. The channel encoder adds redundancy to combat channel impairments and multi-user interference. As discussed in more detail later, our proposed system leverages dynamic channel coding to adapt the transmission rate to the interference perceived at the receiver.

The channel encoder receives a block of length L of uncoded bits. The encoder at node i selects the encoding rate $R_{E,i}$, which represents the number of data bits per encoded bit, among the set $\underline{R}_E = [R_E^1, R_E^2, \cdots, R_E^P]$, where P is the number of different coding rates available and with $R_{E,i}^1 = 1$ (i.e., transmitting uncoded data), and with $R_E^1 > R_E^2 > \cdots >$

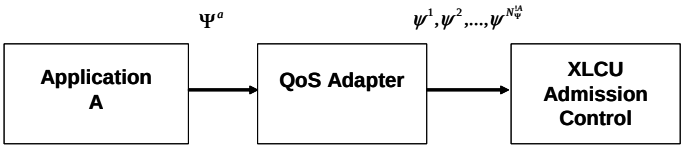


Figure 41: Application data is divided into several flows by a QoS adapter.

R_E^P. Hence, when code R_E^p is selected, with $p \in 1, \cdots, P$, i.e., when $R_{E,i} = R_E^p$, the encoder at i produces a block of coded bits of length L/R_E^p. The set of available codes $\underline{R}_E$ depends on the chosen family of codes, which we will refer to as $\mathcal{C}$. Different families of codes, such as pulse repetition codes or rate-compatible punctured codes, have different performance and different levels of complexity.

5.4.5 Traffic Classes

WMSANs will need to provide support and differentiated service for several different classes of applications. In particular, they will need to provide differentiated service between real-time and delay-tolerant applications, and loss-tolerant and loss-intolerant applications. Moreover, some applications may require a continuous stream of multimedia data for a prolonged period of time (*multimedia streaming*), while some other applications may require event triggered observations obtained in a short time period (*snapshot multimedia content*). The main traffic classes that need to be supported are:

- *Real-time, loss-tolerant, multimedia streams.* This class includes video and audio streams, or multi-level streams composed of video/audio and other scalar data (e.g., temperature readings), as well as metadata associated to the stream, that need to reach a human or automated operator in real time, i.e., within strict delay bounds, and that are however relatively loss tolerant (e.g., video streams can be within a certain level of distortion). Traffic in this class usually has high bandwidth demands.

- *Delay-tolerant, loss-tolerant, multimedia streams.* This class includes multimedia streams that, being intended for storage or subsequent offline processing, do not need

to be delivered within strict delay bounds. However, due to the typically high bandwidth demand of multimedia streams and to limited buffers of multimedia sensors, data in this traffic class needs to be transmitted almost in real-time to avoid excessive losses.

- *Real-time, loss-tolerant, data.* This class may include monitoring data from densely deployed scalar sensors such as light sensors whose monitored phenomenon is characterized by spatial correlation, or loss-tolerant snapshot multimedia data (e.g., images of a phenomenon taken from several multiple viewpoints at the same time). Hence, sensor data has to be received timely but the application is moderately loss-tolerant. The bandwidth demand is usually between low and moderate.

- *Real-time, loss-intolerant data.* This may include data from time-critical monitoring processes such as distributed control applications. The bandwidth demand varies between low and moderate.

- *Delay-tolerant, loss-intolerant, data.* This may include data from critical monitoring processes, with low or moderate bandwidth demand, that require some form of offline post processing.

- *Delay-tolerant, loss-tolerant, data.* This may include environmental data from scalar sensor networks, or non time-critical snapshot multimedia content, with low or moderate bandwidth demand.

The requirements of an application A are described as a set of tuples $\Psi^A = \{\psi^a(\delta^a, \beta^a, \zeta^a),\ a \in 1, \ldots, N_\psi^A\}$. Here, $\psi^a,\ a \in 1, \ldots, N_\psi^A$ represent N_ψ^A different subflows of the flow generated by the application A. For each subflow ψ^a, δ_i^a represents the maximum delay for packets associated with the subflow, β^a represents the required bandwidth, and ζ^a indicates the end-to-end packet error rate that can be sustained by the application. A *QoS Adapter* block, as shown in Fig. 41 can be employed to divide an application layer flow into several

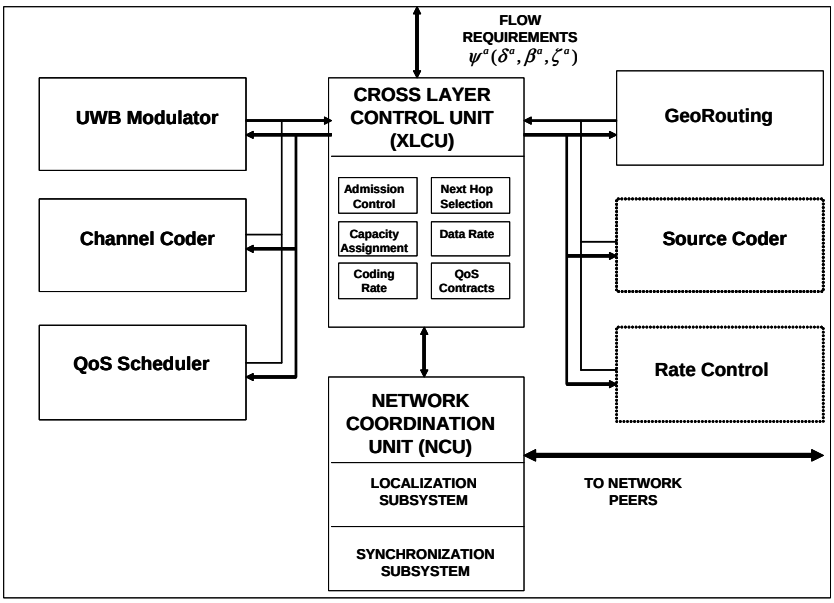


Figure 42: Cross-layer communication architecture.

subflows each with defined characteristics. This allows devising *unequal error protection* schemes in our framework. For example, the new standard for the compression of still images, JPEG2000 [10], incorporates a feature called region of interest (ROI) that may be applicable to visual data sensing. It allows the allocation of greater importance to certain parts of the image, which can then be conveyed in a separate subflow and assigned a different target PER. Moreover, an application can leverage layered multiple description codes with the goal of adapting to heterogeneous clients. Base and enhancement layer descriptions can be associated to different subflows, and undergo the admission control check separately. In the remainder of the paper we consider application flows at the level of subflows, i.e., we assume that a QoS adapter block is available and generates flows of type $\psi^a(\delta^a, \beta^a, \zeta^a)$.

5.5 Design Principles and Cross-layer Controller

In this section, we discuss the design principles that our system is based upon. We assess the benefits of our design in view of the performance objectives and of the characteristics of WMSANs, and describe the cross-layer control architecture of the UWB sensor.

- *Network Layer QoS Support enforced by a cross-layer controller.* The proposed system provides QoS support at the network layer, i.e., it provides packet-level service differentiation in terms of throughput, end-to-end packet error rate, and delay. This is achieved by carefully controlling the interactions between several functionalities at the physical, MAC, and network layer, based on a unified logic that resides on a cross layer controller and that controls the various functionalities by taking decisions based on the parameters advertised by each functional block.

- *UWB Physical/MAC layer.* Our solution is based on an integrated MAC and physical layer solution based on UWB. Similarly to CDMA, TH-IR-UWB allows several transmissions in parallel. Conversely, typical MAC protocols for sensor networks, such as those based on CSMA/CA, require *mutual temporal exclusion* between neighboring transmitters. This allows devising MAC protocols with minimal coordination. While CDMA is usually associated with complex transceivers and cumbersome code assignment protocols, this can be achieved with minimal expense in TH-UWB-IR.

- *Receiver-centric scheduling for QoS Traffic.* One of the major problems in multi-hop wireless environments is that channel and interference vary with the physical location of devices. For this reason, we believe that QoS provisioning becomes easier when based on receiver-centric scheduling. The receiver also controls loss recovery and rate adaptation, thus avoiding feedback overheads and latency, and can be responsive to the dynamics of the wireless link using the information obtained locally.

- *Multi-rate Transmission.* TH-UWB-IR allows changing the data rate at the physical layer, by modifying the *pulse repetition period* (T_f). While this dimension has not been fully explored so far, it is possible to devise adaptive systems that allow modifying the achievable data rate at the physical layer based on the perceived interference and on the required power consumption.

- *Dynamic Channel Coding.* As previously discussed, power control is not beneficial in TH-UWB-IR. Hence, adaptation to interference at the receiver is achieved through dynamic channel coding, which can be seen as an alternative form of power control, as it modulates the energy per bit according to the interference perceived at the receiver.

- *Geographical Forwarding.* Particularly appealing for WMSANs are UWB's positioning capabilities, which are needed in sensor networks to associate physical meaning to the information gathered by sensors. Moreover, knowledge of the position of each network device allows for scalable routing solutions [74]. While angle-of-arrival techniques and signal strength based techniques do not give particular advantages with respect to other transmission techniques, time-based approaches in UWB allow ranging accuracy in the order of centimeters [45]. For this reason, our cross-layer module leverages geographical information to take routing decisions.

- *Hop-by-Hop QoS contracts.* End-to-end QoS requirements are guaranteed by means of local decision. Each single device that participates in the communication process is responsible for locally guaranteeing given performance objectives. The global, end-to-end requirement is thus guaranteed by the joint local decisions of the participating devices.

5.6 Distributed Admission Control Functionality

The operation of the network is based on the concept of *Hop-by-Hop QoS contracts*. As discussed before, the main idea is to try to guarantee end-to-end levels of quality of service by establishing local contracts. Each single device that participates in the communication process is responsible for locally trying to guarantee given performance objectives to devices that are obtaining a service from it. The global, end-to-end objective is thus achieved by the joint local decisions of the participating devices.

The proposed cross-layer controller takes decisions at all layers on the communication stack in order to provide QoS to application flows. The main performance objectives are as follows: i) efficiently exploit the channel, so as to guarantee a high network throughput; ii) minimize the energy consumption; iii) select data paths compliant with the application requirements; iv) guarantee application-level QoS, as described in Section 5.4.5.

Let us assume that a new flow $\psi^a(\delta^a, \beta^a, \zeta^a)$ is generated at node i and requires service. A multi-hop path from i to the destination N needs to be established, with maximum end-to-end delay δ^a, minimum guaranteed bandwidth β^a, and maximum end-to-end packet error rate ζ^a. This is done by establishing *hop-by-hop QoS contracts*. Each device that participates in the communication process is responsible for locally guaranteeing given performance objectives to the devices that it serves, i.e., to the sensors for which it relays data. The global, end-to-end requirement is enforced by the joint local behaviors of the participating devices.

Suppose node i originates a flow $\psi^a(\delta^a, \beta^a, \zeta^a)$. Clearly, the required bandwidth β^a needs to be provided at each hop. As far as delay and packet error rate are concerned, we can afford to allow at each hop a delay and a packet error rate that are proportional to the geographical advance of the packet towards the destination at that hop. For example, if the first hop towards the destination guarantees an advance that equals one third of the total geographical distance towards the destination, then one third of the total allowed end-to-end delay can be allowed to that hop. A similar concept holds for the packet error rate. This can be formalized by considering

$$\delta_{ij}^a = \left(\frac{< d_{ij} >_{iN}}{d_{iN}} \right) \cdot \delta^a, \tag{64}$$

and

$$1 - \left(1 - \zeta_{ij}^a\right)^{\lceil \hat{N}_{ij}^{Hop} \rceil} \leq \zeta^a, \tag{65}$$

which leads to

$$\zeta_{ij}^a \leq 1 - \left(1 - \zeta^a\right)^{\lceil \hat{N}_{ij}^{Hop} \rceil^{-1}} \tag{66}$$

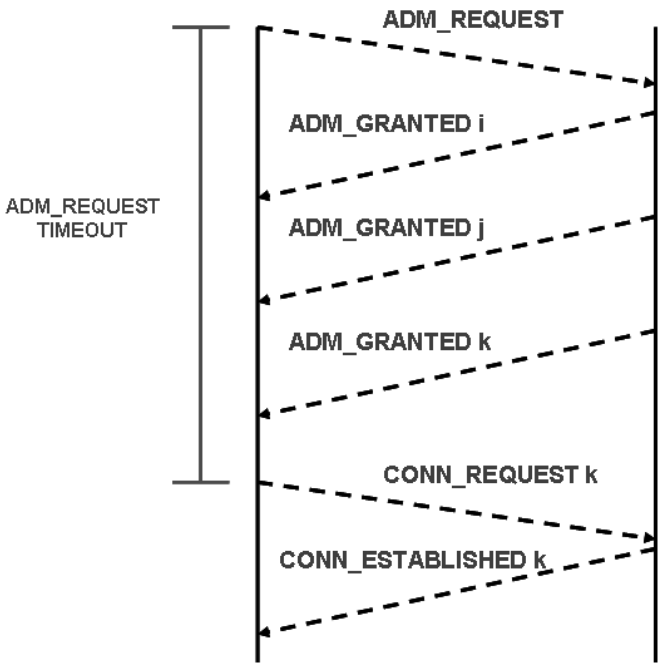


Figure 43: Admission Control and Path Establishment Procedure.

In (64), $< d_{ij} >_{iN}$ (which we refer to as *advance*) is the projection of d_{ij} onto the line connecting node i with the destination, while d_{iN} represents the distance between i and the destination. In (65), by assuming hops of equal length, we express the end-to-end packet error rate and in (66) we derive the minimum requirement ζ_{ij}^{a} for the packet error rate limited to the link between i and j.

The basic operations of the admission control functionalities is shown in Fig. 43. Node i originating the flow broadcasts a short ADM_REQUEST packet, which describes the characteristics of the service required by the set of flows incoming or generated at i. If a neighbor of i i) has positive advance towards the destination actor N with respect to i; ii) is able to provide the requested service with the necessary QoS, it replies with an ADM_GRANTED control packet, which also includes a value quantifying the level of service provided by the device. This can reflect different performance metrics such as energy consumption, the available level of batteries, etc. Hence, node i receives an ADM_GRANTED packet from all neighbors able to guarantee the service. Among these, the optimal relay node

is selected as the node according to an optimization criterion described in the following. The node requesting the service will then send a `CONTRACT_REQUEST` packet to the selected node, that will reply with a `CONTRACT_ESTABLISHED` message that creates the connection. Recursively, the end-to-end path will be established until the destination actor is reached. If no `ADM_GRANTED` message is received, the procedure is aborted and a `CONTRACT_RESCINDED` message is sent to the upstream node, which will blacklist the downstream node and run the admission control procedure again.

Formally, a local optimization problem is distributively solved by the devices involved, the solution of which determines the optimal data path. Let us introduce the following:

- $E^{pulse} = 2 \cdot E^{pulse}_{elec} + P^{TX} \cdot T_{f,i}\,[\mathrm{J/pulse}]$ accounts for the energy to transmit one pulse from node i to node j, where E^{pulse}_{elec} is the energy per pulse needed by transmitter electronics and digital processing; $P^{TX}\,[\mathrm{W}]$ and $T_{f,i}\,[\mathrm{s}]$ are the average transmitted power and the frame length, respectively.

- $\hat{N}^{TX,a}_{ij}$ is the average number of transmissions of a packet from flow a for the packet to be correctly decoded at receiver j. The actual value $\hat{N}^{TX,a}_{ij} = \frac{1}{1-PER^{\mathcal{C},L}_{ij}}$ depends on the interference perceived at the receiver, on the coding scheme $\mathcal{C}$ adopted, and on the packet size L.

- $\hat{N}^{Hop}_{ij} = \max\left(\frac{d_{iN}}{<d_{ij}>_{iN}}, 1\right)$ is the estimated number of hops from node i to the destination N when j is selected as next hop.

- $\mathcal{S}_i$ is the *neighbor set* of node i, while $\mathcal{P}^N_i$ is the *positive advance set*, of i, i.e., $j \in \mathcal{P}^N_i$ iff $d_{jN} < d_{iN}$.

- $\mathcal{F}_i$ is the set of incoming or generated flows at node i.

- The bandwidth requirement β^a of application a can be expressed as $\beta^a = R^a_{0,i} \cdot R^a_{E,i}$, where $R^a_{0,i}\,[\mathrm{pulses/s}]$ represents the raw pulse rate for application a required to achieve the rate β^a, when a coding rate $R^a_{E,i}$ is used.

- $\beta^{tot} = \sum_{a \in \mathcal{F}_i} \beta^a$ represents the total bandwidth requirement, in bits/s, for flows incoming or generated at i.

$\mathbf{P^{dist}}$: Distributed Admission Control, Routing and Channel Coding Problem

$$\begin{aligned} Given: &\quad i,\ N,\ \mathcal{S}_i,\ \mathcal{P}_i^N,\ E^{pulse},\ \mathcal{F}_i \\ Find: &\quad j^* \in \mathcal{S}_i \cap \mathcal{P}_i^N,\ R_{E,i}^a\ \forall a \in \mathcal{F}_i \\ Minimize: &\quad E_{(i,j)}^{bit} = \frac{1}{\beta^{tot}} \sum_{a \in \mathcal{F}(i)} \frac{E^{pulse} \cdot \hat{N}_{ij}^{TX,a} \cdot \hat{N}_{ij}^{Hop} \cdot \beta^a}{R_{E,j}^a} \\ Subject\ to: & \end{aligned} \tag{67}$$

Packet Error Rate:

$$R_{E,i}^a \leq \min\Big(\frac{E_i^{(r)}}{\gamma_{\mathcal{C},j}^a(\zeta_{ij}^a)[\eta_j + \frac{\sigma^2}{T_{f,j}} \sum_{k \in \mathcal{F}(i), k \neq i} E_k^{(r)}]}, 1\Big),\ \forall a \in \mathcal{F}_i; \tag{68}$$

Rate Admission Control:

$$\sum_{a \in \mathcal{F}_j} \frac{\beta^a}{R_{E,\mathcal{N}_j}^a(\gamma_{\mathcal{C},\mathcal{N}_j}^a(\zeta_{j\mathcal{N}_j}^a)} + \\ + \sum_{a \in \mathcal{F}_j} \frac{\beta^a}{R_{E,\mathcal{U}_j^a}^a(\gamma_{\mathcal{C},j}^a(\zeta_{\mathcal{U}_j^a j}^a))} + \frac{R_j^{sched,up}}{R_{E,j}^{sched}} + \frac{R_j^{sched,down}}{R_{E,j}^{sched}} \leq R_{0,j} \tag{69}$$

Delay Admission Control:

$$\sum_{a \in \mathcal{F}_i} \frac{LR_{0,j}}{R_{E,\mathcal{N}_j}^a} + T^{sched,up} + T^{sched,down} + \\ + \sum_{a \in \mathcal{F}_i} L\Big(1 + \frac{b_j^a}{\phi_j^a}\Big) \cdot \frac{1}{R_{0,j} R_{E,j}^a} + \frac{L}{R_{0,j} R_{E,j}^a} \leq \delta_{ij}^a,\ \forall a \in \mathcal{F}_i. \tag{70}$$

According to the proposed routing rule, i will select j^* as its best next hop iff

$$j^* = argmin_{j \in \mathcal{S}_i \cap \mathcal{P}_i^N} E_{(i,j)}^{bit}, \tag{71}$$

where $E_{(i,j)}^{bit}$ represents the minimum average energy required to successfully transmit a payload bit from node i to the destination, given the interference at j, when i selects j as

next hop. This link metric, objective function (67) in $\mathbf{P^{dist}}$, takes into account the average number of packet transmissions $\hat{N}_{ij}^{TX,a}$ associated with link (i,j) and flow a. Moreover, it accounts for the average hop-path length $(\hat{N}_{ij}^{Hop})$ from node i to the destination when j is selected as next hop, by assuming that the following hops will guarantee the same advance towards the destination. While this is a simple way to estimate the number of hops towards the destination, i) it does not incur any signaling overhead; ii) its accuracy increases as the density increases; iii) its accuracy increases as the distance to the destination decreases.

The solution of the problem can be interpreted as decoupling $\mathbf{P^{dist}}$ into three sub-problems: first, at each feasible next hop (neighbor with positive advance), find, if it exists, the minimum coding rate for i to meet the local PER requirement ζ_{ij}^a for each flow a in $\mathcal{F}_i$ (constraint (68)); second, check if given the required coding rates, node j has sufficient bandwidth (constraint 69) and can provide service to the flows with the required delay (constraint 84). In practice, this first three steps are performed at each node receiving the `ADM_REQUEST` packet. Finally, among the nodes that have granted admission, node i picks the node j^* with minimal link metric. Note that node i solves, by enumeration, a low-complexity problem.

Constraint (68) defines the minimum coding rate $R_{E,i}^a$ required at node i to send a packet towards neighbor j in order to guarantee a minimum signal-to-noise-plus-interference (SINR) ratio $\gamma_{\mathcal{C},j}^a(\zeta_{ij}^a)$ at j, i.e., the minimum SINR needed to guarantee a packet error rate ζ_{ij}^a, given the interference generated by the other UWB signals at j (denominator of the expression), as derived in Section 5.7. Constraint (69) checks if node j has enough bandwidth to satisfy the request, i.e., if the sum of the raw physical data rates of the incoming flows at j (first term in the sum) plus the outgoing flows (second term) plus the data rate to transmit control packets to determine schedules in the upstream and downstream directions are lower than the raw physical data rate $R_{0,j}$ at j. Here, $\mathcal{N}_j$ represents the next hop of j while $\mathcal{U}_j^a$ represents the upstream node of j for flow a. Finally, constraint (84) checks if j is able to provide service with the required delay. The bound is derived by assuming a wireless fair

service approach [66], and extending it for a multi-rate, multi-hop environment with dynamic channel coding with concurrent UWB transmissions, as further discussed in Section 5.7. Alternative dynamic scheduling techniques can be also included in the framework and will be object of future work.

5.7 Scheduling and Rate Assignment

In this section, we describe how our cross-layer module schedules data packets and assigns data rates to different flows based on the application requirements.

We consider a sensor node i, receiver of several application flows. We further assume that

$$\psi^a(\delta^a, \beta^a, \zeta^a) \Longrightarrow (\delta_i^a, \beta_i^a, \zeta_i^a) = \Theta(\psi^a, \mathbf{x}_i, \mathbf{x}_N),\ \forall a \in \mathcal{F}(i), \tag{72}$$

i.e., the requirements of each application flow a that belongs to the set $\mathcal{F}(i)$ of *incoming flows in* i induce requirements $(\delta_i^a, \beta_i^a, \zeta_i^a)$ associated to i and the application a, that depend on the characteristics of the application and on the geographical position of i and of the destination actor N.

5.7.1 Rate Assignment

The *Signal to Interference plus noise ratio* at node i ($SINR_i$) can be expressed as [30]:

$$SINR_i = \frac{P_i g_{ii}}{R_i[\eta_i + \sigma'^2 T_{f,i} \sum_{j \in \mathcal{F}(i), j \neq i} P_j g_{ji}]}\ \ i = 1, \cdots, N \tag{73}$$

where $P_i\,[\mathrm{W}]$ represents the transmitted power, g_{ij} represents the path loss from the from the transmitter on the i^{th} link to the receiver on the j^{th} link, $R_i\,[\mathrm{bits/s}]$ represents the data rate on the i^{th} link, $\eta_i\,[\mathrm{V^2s}]$, and represents the background noise energy plus interference from other non UWB systems. Moreover, $T_{f,i}\,[\mathrm{s}]$ represents the length of the physical layer frame on the i^{th} link, while σ' is an a-dimensional parameter that depends on the shape of

the monocycle and on the frame length. For our purpose, it turns useful to rewrite $\sigma^{'2}$ as

$$\sigma^{'2} = \frac{1}{T_f} \cdot \frac{\int_{-\infty}^{+\infty} \left[\int_{-\infty}^{+\infty} p_{rec}(x-\tau)v(x)dx \right]^2 d\tau}{\int_{-\infty}^{+\infty} p_{rec}(x-\delta)v(x)dx} = \frac{1}{T_f} \cdot \sigma^2. \tag{74}$$

In the above expression, p_{rec} represents the received impulse shape and $v(t) = p_{rec}(t) - p_{rec}(t-\delta)$ represents the *correlator's template signal* at the receiver. While the newly defined parameter $\sigma\,[\mathrm{s}^{1/2}]$ depends only on the shape of the monocycle and on the type of modulation used, we make the dependence of $\sigma^{'2}$ on the frame length T_f explicit. By defining $P_j^{rec} = P_j g_{ji}$ we can write:

$$SINR_i = \frac{P_i^{(r)}}{R_i[\eta_i + \frac{\sigma^2}{T_{f,i}} T_{f,i} \sum_{j \in \mathcal{F}(i), j \neq i} P_j^{(r)}]} \quad =$$

$$= \frac{E_i^{(r)}}{T_{f,i} R_i[\eta_i + \frac{\sigma^2}{T_{f,i}} \sum_{j \in \mathcal{F}(i), j \neq i} E_j^{(r)}]}, \tag{75}$$

where $E_j^{rec} = P_j^{rec} T_{f,i}$ represents the received energy per pulse from the j^{th} transmitter.

Now, given a flow $\psi^a(\delta^a, \beta^a, \zeta^a)$, the end-to-end requirement on the maximum end-to-end packet error rate ζ^a imposes certain packet error rate to the receiver i $\zeta_i^a = \Theta(\psi^a, \mathbf{x}_i, \mathbf{x}_N)$. Hence, we need to have

$$SINR_i^a \geq \gamma_{\mathcal{C},i}^a(\zeta_i^a). \tag{76}$$

where $\gamma_{\mathcal{C},i}^a(\zeta_i^a)$ is the SINR threshold that guarantees the packet error rate ζ_i^a required by flow a at node i, given the chosen family of error correcting codes $\mathcal{C}$. The expression in (76) can be rewritten as

$$R_i^a \leq \min\Big(\frac{E_i^{(r)}}{T_{f,i} \gamma_{\mathcal{C},i}^a(\zeta_i^a)[\eta_i + \frac{\sigma^2}{T_{f,i}} \sum_{j \in \mathcal{F}(i), j \neq i} E_j^{(r)}]}, \frac{1}{T_{f,i}} \Big), \tag{77}$$

and by substituting $R_i^a = R_{E,i}^a R_{0,i}$

$$R_{E,i}^a \leq \min\Big(\frac{E_i^{(r)}}{\gamma_{\mathcal{C},i}^a(\zeta_i^a)[\eta_i + \frac{\sigma^2}{T_{f,i}} \sum_{j \in \mathcal{F}(i), j \neq i} E_j^{(r)}]}, 1 \Big). \tag{78}$$

Hence, the optimal coding rate for flow a is selected as

$$R_{E,i}^a = \max_{1 \leq p \leq P} R_E^p \; s.t. \; (78) \; holds \tag{79}$$

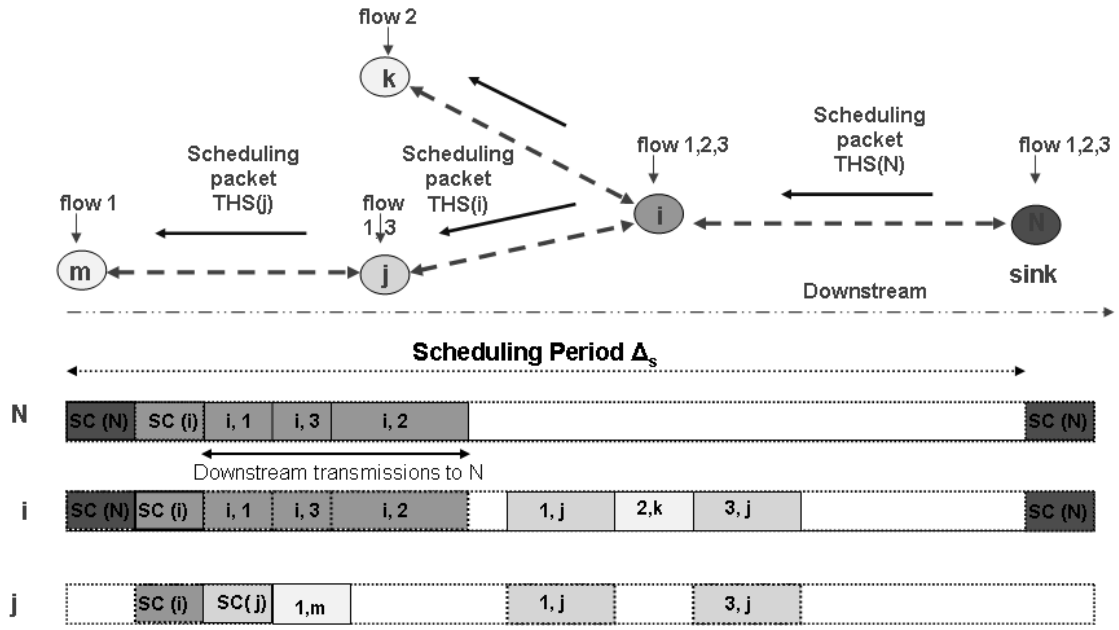


Figure 44: Scheduling of data packets.

5.7.2 Receiver-centric Scheduling

For unicast transmissions, a pseudo-random time hopping sequence $THS(j)$ is generated using the identity of the receiver j as the seed of the random number generator, while for multicast transmissions the time hopping sequence $THS(i)$ is generated based on the identity of the transmitter i. Coordination of medium access is still needed to:

1. **Prevent collisions at the receiver.** When a device i is receiving data from a device j, no other device should transmit data intended for i (i.e., using $THS(i)$) simultaneously, as we assume that i is endowed with a simple single-user receiver.

2. **Avoid idle listening.** Each device should be tuned to the wireless channel only when incoming transmissions for itself are occurring, i.e., each device should consume energy only when actually receiving data.

3. **Avoid wasteful transmissions.** When a device i is transmitting data to j, j's receiver must be tuned to $THS(j)$ to listen for incoming transmissions.

Our objective is therefore to devise a medium sharing policy that achieves the above objectives with simple coordination. Our solution is illustrated in Fig. 44. Each device is responsible for scheduling transmissions of data packets from its upstream nodes, i.e., the devices it is offering a service to, i.e., $\forall u \in \mathcal{F}_i$. Device i prepares a `SCHEDULE` packet, that is transmitted at periodic intervals Δ_s. The scheduling period Δ_s is known to all network devices. The phase Φ_s^i is communicated by i to its upstream nodes in the `CONTRACT_ESTABLISHED` message. The `SCHEDULE` packet is broadcast by i and all its upstream nodes receive it by periodically tuning their UWB receiver to $THS(i)$. A *schedule* is a vector of *appointments*, i.e., tuples $(a, u, t_k^a, R_{E,u}^a)$, where a represents an application flow, u represents a node, $u \in \mathcal{F}_i$, t_k^a represents the starting time for transmission of the k^{th} packet from flow a at u, and $R_{E,u}^a$ represents the required coding rate. By sending an appointment $(a, u, t_k^a, R_{E,u}^a)$, node i commits to receiving a packet from u from flow a starting at time t_k^a for a time equal to $L/(R_0 R_E^a)$, where $L\,[\mathrm{bit}]$ is the packet length. Nodes in $\mathcal{F}_i$ transmit a scheduling packet for their upstream nodes, if they have any, immediately after receiving the scheduling packet from i. Hence, when preparing schedules for their upstream nodes, they can consider previous commitments with their downstream node. In this way, the downstream (closer to sink) node of each node has priority in deciding appointments. Hence, conflict-free scheduling can be achieved in a very simple way. This is only paid in terms of flexibility, as all incoming flows have to be transmitted downstream through the same next-hop, i.e., multi-path routing does not fit in this framework. However, this is a price worth paying for the simplicity achieved.

We determine the actual scheduling of packets from upstream nodes based on a procedure inspired by the wireless fair scheduling (WFS) paradigm. WFS [20] is a family of solutions designed to guarantee delay-bounded and throughput-guaranteed access in single-hop, single-rate wireless packet networks (i.e., cellular networks). Most of these solutions are based on wireless adaptations of the packetized version of the Generalized Processor Sharing (GPS) paradigm [84]. We consider a wireless fair service approach [66] and extend

it for a UWB multi-rate, multi-hop environment.

Consider a node i, relayer of a set of incoming flows from its upstream nodes $\mathcal{F}_i$. We denote the next hop of i towards the sink N as $\mathcal{N}_i^a$. The k^{th} incoming packet of the a^{th} flow $p_{i,k}^a$ is start-tagged as

$$S(p_{i,k}^a) = \max\{S(p_{i,k-1}^a) + \frac{L_{k-1}^a}{b_i^a},\ A(p_{i,k}^a)\}, \tag{80}$$

where L_{k-1}^a is the length of packet $k-1$ for flow a, b_i^a is called the *bandwidth coefficient*, and $A(p_{i,k}^a)$ represents the arrival time of the packet. The finish tag is set as

$$F(p_{i,k}^a) = S(p_{i,k}^a) + \frac{L_k^a}{\phi_i^a}, \tag{81}$$

where ϕ_i^a is called the *delay coefficient*. At each step, the scheduler transmits first the packet with the lowest finish time. The bandwidth requirement β^a of flow a can be expressed as $\beta^a = R_{0,i}^a \cdot R_{E,i}^a$. Hence, we define the *bandwidth coefficient* b_i^a as

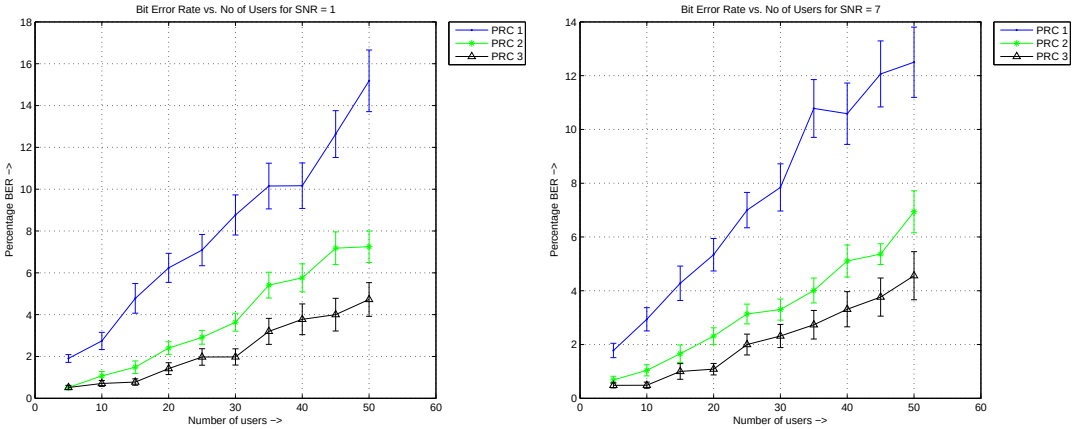
$$b_i^a = \frac{R_{0,i}^a}{R_{0,i}^{TOT,IN}} = \frac{\frac{\beta_i^a}{R_{E,i}^a}}{\sum_{b\in\mathcal{F}_i}\frac{\beta_i^b}{R_{E,i}^b}}. \tag{82}$$

We define the *delay coefficients* ϕ_i^a as

$$\phi_i^a = 1 - \frac{\delta_i^a}{\sum_{b\in\mathcal{F}_i}\delta_i^b} = \frac{\sum_{b\in\mathcal{F}_i, b\neq a}\delta_i^b}{\sum_{b\in\mathcal{F}_i}\delta_i^b}. \tag{83}$$

Note that the value of the bandwidth and delay coefficients, which are fundamental parameters of the schedulers, are constantly updated by the XLCU to reflect the interference perceived at the receiver, changes in paths, and the application requirements so as to assign transmission opportunities that reflect the requirements of the flows being served. With the above definitions, the new queue delay $D_j^{a,new}$ of the head of line packet of flow a is bounded by

$$D_j^{a,new} \leq \sum_{a\in\mathcal{F}_i}\frac{LR_{0,j}}{R_{E,\mathcal{N}_j}^a} + T^{sched,up} + T^{sched,down} + \\ + \sum_{a\in\mathcal{F}_i} L\Big(1 + \frac{b_j^a}{\phi_j^a}\Big)\cdot\frac{1}{R_{0,j}R_{E,j}^a} + \frac{L}{R_{0,j}R_{E,j}^a}. \tag{84}$$



(a) (b)

Figure 45: Bit Error Rate with increasing number of users, for different Pulse Repetition Codes, for SNR=1dB (a) and SNR=7dB (b).

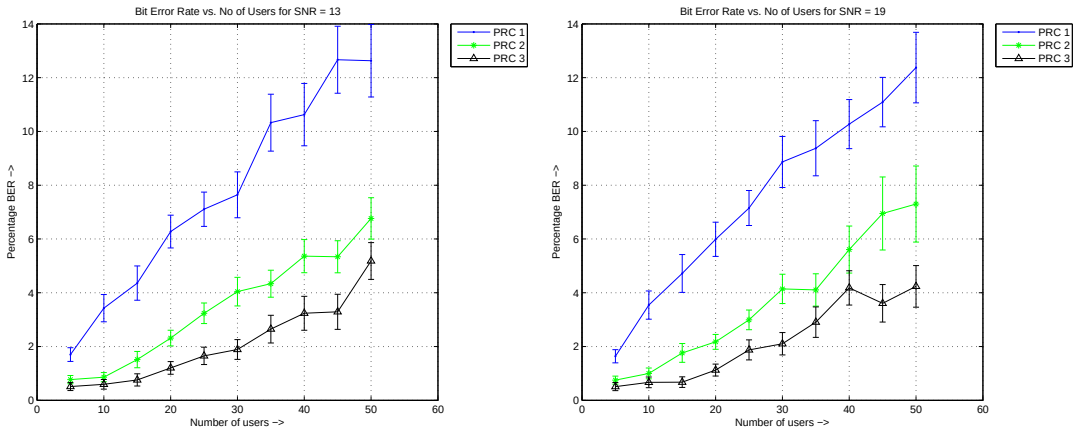
The above bound, where $T^{sched,up}$ and $T^{sched,down}$ represent the time needed to transmit the schedule packets upstream and downstream, respectively, is used by the admission control procedure, as explained in Section 5.6.

5.8 Performance Evaluation

In this section we discuss performance evaluation results for the solution proposed in this chapter for cross layer QoS support in wireless multimedia sensor and actor networks. To assess the performance of the proposed solution, we have developed two different simulation tools, i.e., a physical layer simulator of the low-level physical layer TH-IR-UWB communication architecture in Matlab, and a discrete-event object-oriented network simulator, which was developed in Java.

5.8.1 Physical Layer Simulations

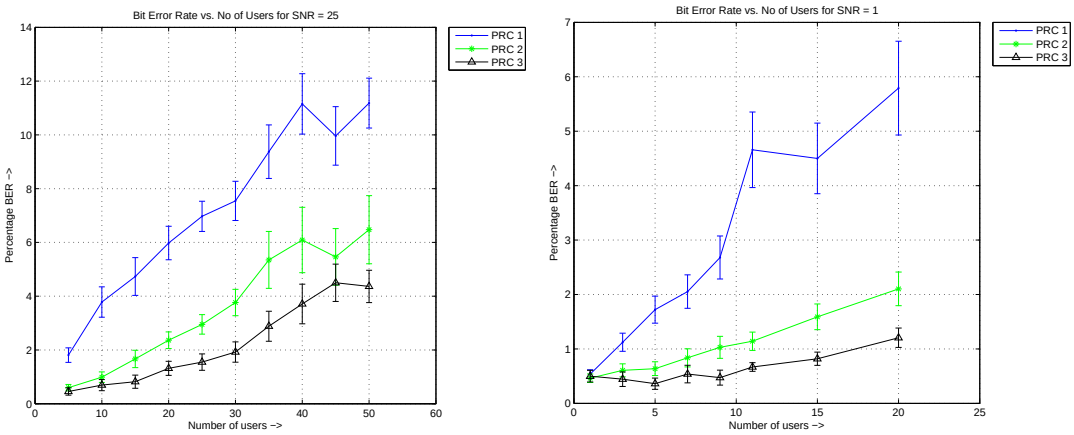
The physical layer simulator models the TH-IR-UWB communication process in detail, as described in Section 5.4.2. The simulator models generation of Gaussian pulses, transmission on the multi-path affected UWB channel, interference from concurrent transmitters, and reception based on a correlation receiver, affected by multi-user interference and white Gaussian noise. Basically, a pulse is generated, and transmitted on the multipath channel.



(a) (b)

Figure 46: Bit Error Rate with increasing number of users, for different Pulse Repetition Codes, for SNR=13dB (a) and SNR=19dB (b).

The received signal is calculated by multiplying the Fast Fourier Transform (FFT) of the signal with the FFT of the impulse response of the channel, and by performing the inverse FFT. The signal is then detected through a correlation receiver, which assumes perfect knowledge of the channel. An extensive simulation campaign has been conducted with this simulator to assess the performance of TH-IR-UWB. The results from this campaign provided us with an exhaustive set of results, expressed in terms of bit error rate, which have been then used as input for the packet-level simulator developed in Java. All figures shown in this section report 95% confidence intervals, and for each point in the figure we repeat simulations with a block of $1000\,\mathrm{bits}$ until the relative error, i.e., the ratio between the single-sided confidence interval and the estimated value, is below 15%. Figure 45 shows the bit error rate (BER) of TH-IR-UWB communications with increasing number of users, from 1 to 50, with different Pulse Repetition Codes, for SNR=1dB (a) and SNR=7dB (b). The SNR is calculated by adding white Gaussian noise at the receiver. As expected, the bit error rate increases with increasing number of users, and decreases with higher pulse repetition codes (PRC), i.e., by conveying the same information bit with a higher number of pulses. As can be noted, while the PRC code used consistently affects the attainable BER, the signal-to-noise ratio at the receiver, although it does have an impact, does not influence



(a) (b)

Figure 47: Bit Error Rate with increasing number of users, for different Pulse Repetition Codes, for SNR=25dB (a) and SNR=1dB with up to 20 users (b).

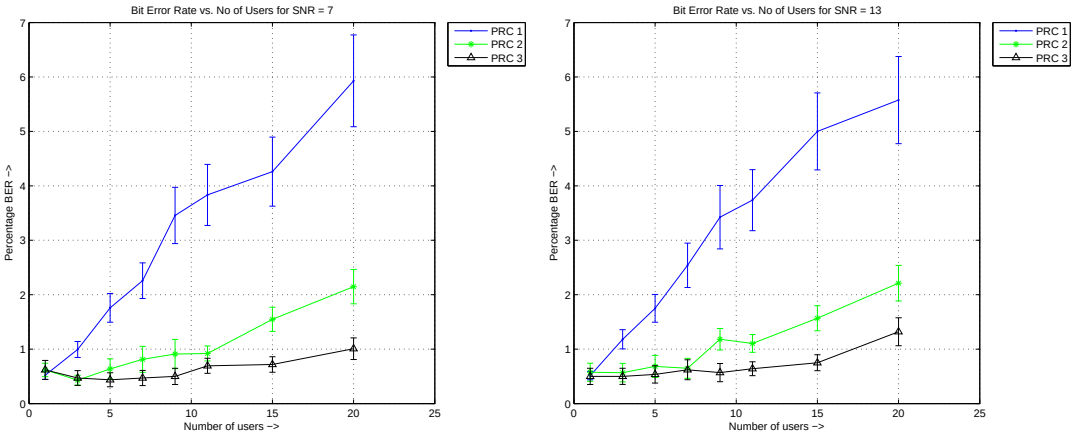
the achievable bit error rate to the same extent that the PRC or the number of users does.

Figures 46 and 47(a) show the Bit Error Rate with increasing number of users (from 1 to 50), for different Pulse Repetition Codes, for SNR=13dB, SNR=19dB, and SNR=25dB, respectively. Figures 47(b), 48(a), 48(b), 49(a), 49(b) show the Bit Error Rate with increasing number of users (from 1 to 20, with a denser sampling), for different Pulse Repetition Codes, for SNR=1dB, SNR=7dB, SNR=13dB, SNR=19dB, and SNR=25dB, respectively.

5.8.2 Network Simulations

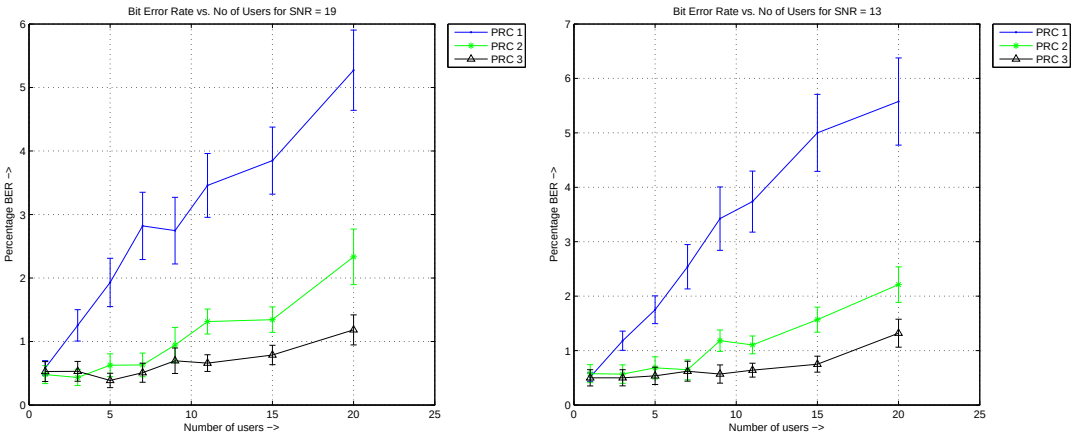
Network simulations were performed by developing a discrete-event object-oriented network simulator in Java. The simulator models all aspects of the communication architecture described in this chapter. The greatest challenge in its development was to maintain a modular architecture, which reduces the burden of design and allows for easier and more efficient debugging, while allowing modeling of the cross-layer interactions that constitute the hearth of the proposed architecture.

Results from the Matlab physical layer simulator contributed to a module that accurately models UWB propagation, interference, and energy consumption. Figures 50 and 51 are from a simple scenario such as that represented in Fig. 44, where there are two



(a) (b)

Figure 48: Bit Error Rate with increasing number of users, for different Pulse Repetition Codes, for SNR=7dB (a) and SNR=13dB (b).



(a) (b)

Figure 49: Bit Error Rate with increasing number of users, for different Pulse Repetition Codes, for SNR=19dB (a) and SNR=25dB (b).

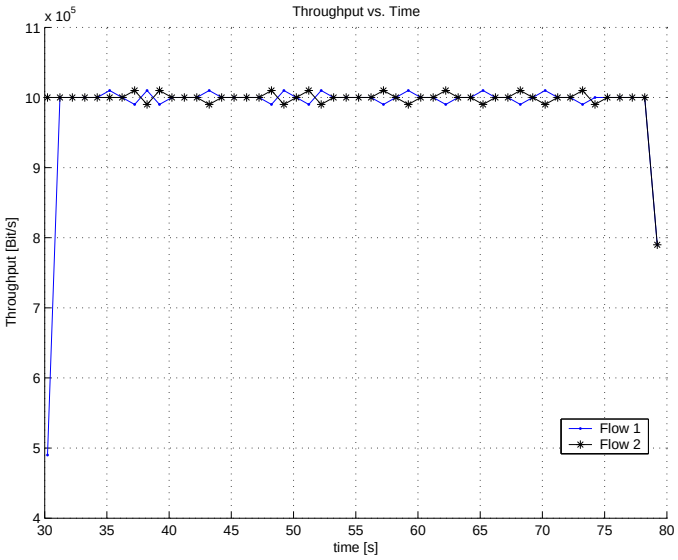


Figure 50: Scenario 1. Throughput vs. Time for two different flows.

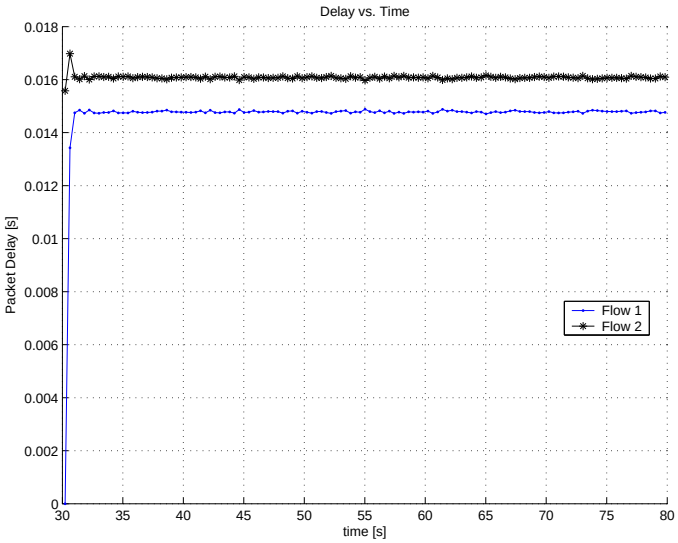


Figure 51: Scenario 1. Delay vs. Time for two different flows.

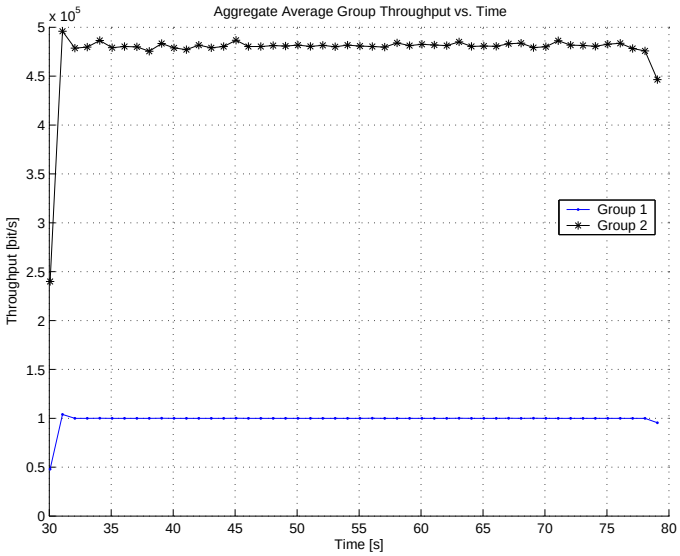


Figure 52: Scenario 2. Aggregate Average Group Throughput vs. Time

sources, both generating traffic requiring 1Mb/s, and with equal loose delay requirements. Figure 50 shows that both sources receive in average the required service, and Fig. 51 shows that the end-to-end packet delivery delays are very short (around $15\,\mathrm{ms}$), do not fluctuate (low jitter), and are comparable for the two sources. The difference in the delays is only caused by unsynchronized packet generation time at the sources.

The second considered scenario consists of a $200\,\mathrm{m}\,\mathrm{x}\,200\,\mathrm{m}$ terrain where 49 nodes are deployed in a grid structure. There are 2 groups of 12 constant bit rate sources, one located over the lower left corner of the grid, and the other one at the upper left corner. Flows in group 1 require $100\,\mathrm{kbit/s}$ bandwidth, $100\,\mathrm{ms}$ end-to-end delay, and 0% PER. Flows in group 2 have higher bandwidth demand ($500\,\mathrm{kbit/s}$), $100\,\mathrm{ms}$ end-to-end delay and can admit 10% PER. The sink is located in the middle of the right side of the square. Figure 52 shows the average aggregate throughput for sources belonging to the two groups. Sources in group 1 have a throughput of exactly $100\,\mathrm{kbit/s}$, while sources in group 2 show an average throughput of about $480\,\mathrm{kbit/s}$, as some packets are lost. Figure 53 shows a bar plot of the packets generated, received and lost per flow. While flows in group 1 do not lose packets, flows in group 2 lose approximately 4% of the packets, which is still below the

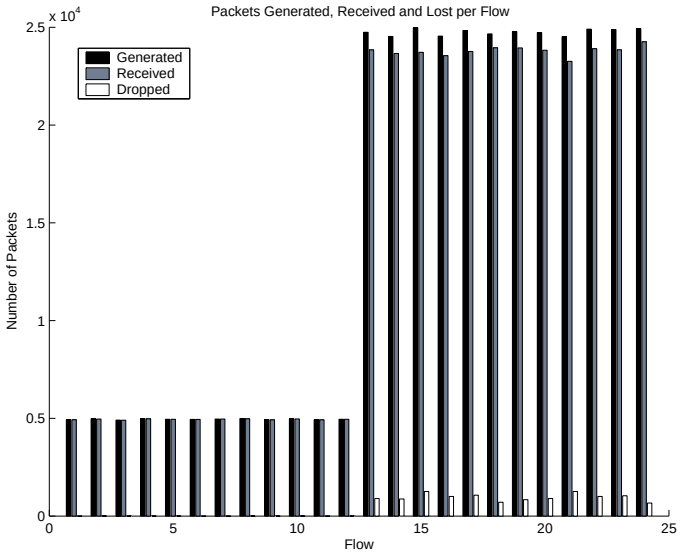


Figure 53: Scenario 2. Packets Generated, Received and Lost per Flow.

application requirement. Note that this is achieved with more redundant pulse repetition codes for nodes in group 1. In average, each bit for a flow in group 1 is sent with a coding rate of $1/3$, while the coding rate in group 2 is in average very close to 1. This directly translates into a consistently higher energy consumption. More complex coding schemes can achieve a better energy efficiency at the expense of complexity.

Figure 54 shows a comparison between the delays of the two groups with time. The aggregate average end-to-end delays of the two groups are well below the threshold end-to-end delay. The higher delays shown by flows in group 1 are very limited in absolute value (around $10\,\mathrm{ms}$) and are caused by the lower coding rate employed by sources in this group, which lead to higher transmission time. Finally, Fig. 55 shows a bar plot of the average end-to-end delay and its variance. The differences in delays between flows in the same groups are very limited between different flows, which demonstrates the basic fairness of the system, and the variance of the delay is also limited, which shows that under normal circumstances the system leads to much more limited jitter as compared to CSMA/CA based systems.

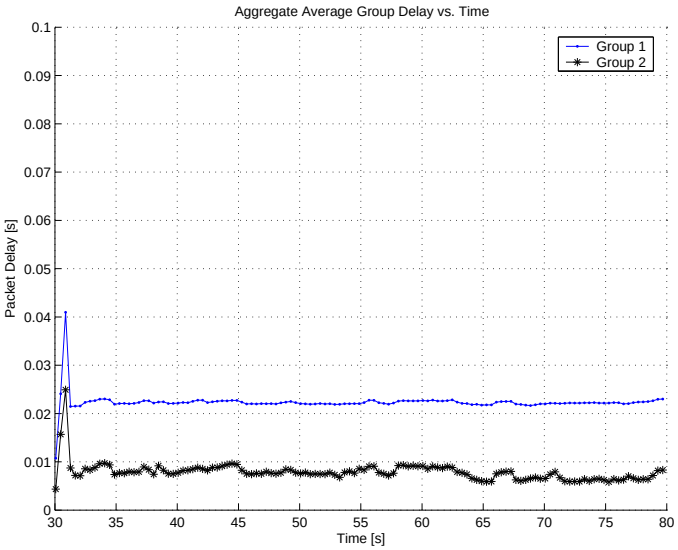


Figure 54: Scenario 2. Aggregate Average Group Delay vs. Time.

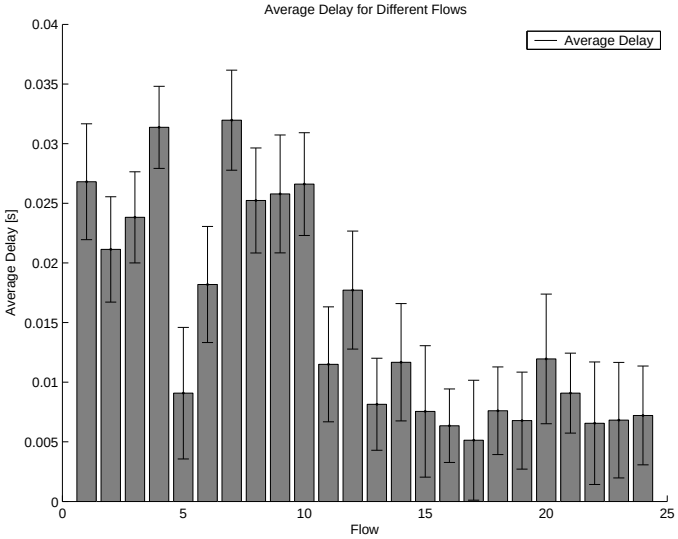


Figure 55: Scenario 2. Average Delay for Different Flows.

CHAPTER VI

CONCLUSIONS

In this thesis, we have discussed several communication and coordination problems in the new domain of Wireless Sensor and Actor Networks (WSANs), for data-centric and multimedia application scenarios. In particular, we have dealt with general resource and task allocation problems in these networks, which we have modeled as mathematical problems, with the objective of developing distributed solutions that provide near optimal performance.

In Chapter 2, we have presented a coordination framework for static WSANs, and discussed the sensor-actor and actor-actor coordination problems. We developed an optimal solution for the sensor-actor coordination based on an event-driven partitioning paradigm, and formulated it as an integer linear program. We also proposed DEPR, a distributed protocol for sensor-actor coordination that includes an adaptive mechanism to trade off energy consumption for delay when the event data has to be delivered to the actors within predetermined latency bounds. For the actor-actor coordination, an optimization model was defined for a class of coordination problems in which the area to be acted upon is optimally split among different actors. The problem was formulated as a mixed integer non-linear problem, and an auction-based localized solution of the problem was presented. Future work will be focused on extending the proposed solutions, i.e., generalizing the framework to capture different application scenarios, e.g., with multiple concurrent events, and to implement the proposed solutions on a hardware testbed.

In Chapter 3, we have outlined the challenges for coordination and communication in Wireless Sensor and Actor Networks (WSANs) with mobile actors, and presented effective solutions for the sensor-actor and actor-actor coordination problems. First, we proposed

a proactive location management scheme to handle the mobility of actors with minimal energy expenditure for sensors. Then, an energy efficient communication solution was derived for sensor-actor communication based on geographical routing. We showed how to control the delay of the data-delivery process based on power control, and how to deal with network congestion by forcing multiple actors to share the traffic generated in the event area. Finally, a model for actor-actor coordination was introduced that coordinates motion based on the characteristics of the event. The solutions proposed enable several new application scenarios for distributed systems of wireless sensors and mobile actors.

In Chapter 4, we discussed the state of the art of research on Wireless Multimedia Sensor and Actor Networks (WMSANs), and outlined the main research challenges. Algorithms, protocols, and hardware for the development of WMSANs were surveyed, and open research issues discussed in detail. We classified currently off-the-shelf hardware as well as available research prototypes for WMSANs. Furthermore, we discussed existing solutions and open research issues at different layers of the communication stack, along with possible cross-layer synergies and optimizations. We believe that this research area will attract the attention of many researchers and that it will push one step further our ability to observe the physical environment and interact with it.

In Chapter 5, we have finally described our efforts in the design of a cross-layer communication architecture to provide quality of service support to multimedia flows for wireless multimedia sensor and actor networks. The design is based on time hopping impulse radio ultra wide band communications, which is a promising physical layer technique for low-power high rate wireless communications. The communication architecture is based on an innovative design and aims at providing differentiation for different flows in the domains of throughput, delay, reliability, and energy consumption, based on a modular cross-layer controller that performs admission control, routing, scheduling, bandwidth assignment and coding with the objective of satisfying the requirements of the flows involved. Performance evaluations shows that the architecture is a promising solution to satisfy the performance

objectives of wireless multimedia sensor and actor networks.

We plan to extend our work in several directions. On the short term, our future work will be aimed at extending the cross-layer architecture to include adaptive modulation, and to provide differentiated service for peak data rate and average data rate. Moreover, we will evaluate in the framework the effect of refined ARQ strategies. Finally, we will incorporate in the design adaptive source coding for multimedia traffic and end-to-end transport layer solutions.

As a long-term objective, we plan to build on the lessons learnt from research on these topics to work toward developing a unified theory of sensing, communication, and control, that will allow predicting the performance of these complex systems, and be the basis to develop design tools that will help companies realize this systems according to user requirements.

REFERENCES

[1] "Acroname GARCIA Robotic Platform." http://www.acroname.com/garcia/garcia.html.

[2] "Cplex solver." http://www.cplex.com.

[3] "Crossbow." http://www.xbow.com.

[4] "Crossbow MICA2 Mote Specifications." http://www.xbow.com.

[5] "Crossbow MICAz Mote Specifications." http://www.xbow.com.

[6] "Crossbow TelosB Mote Specifications." http://www.xbow.com.

[7] "The j-sim simulator." http://www.j-sim.org/.

[8] "Revision of Part 15 of the Commission's Rules Regarding Ultra-Wideband Transmission Systems." First note and Order, Federal Communications Commission, ET-Docket 98-153, Adopted February 14, 2002, released April 22, 2002.

[9] "The Stargate Platform." http://www.xbow.com/Products/Xscale.htm.

[10] "JPEG2000 Requirements and Profiles." ISO/IEC JTC1/SC29/WG1 N1271, March 1999.

[11] AHUJA, R. K., MAGNANTI, T. L., and ORLIN, J. B., *Network Flows: Theory, Algorithms, and Applications*. Englewood Cliffs, New Jersey: Prentice Hall, February 1993.

[12] AKAN, O. and AKYILDIZ, I. F., "Event-to-Sink Reliable Transport in Wireless Sensor Networks," *IEEE/ACM Transactions on Networking*, vol. 13, pp. 1003–1017, October 2005.

[13] AKYILDIZ, I. F. and KASIMOGLU, I. H., "Wireless sensor and actor networks: Research challenges," *Ad Hoc Networks (Elsevier)*, vol. 2, pp. 351–367, October 2004.

[14] AKYILDIZ, I. F., MELODIA, T., and CHOWDURY, K. R., "A Survey on Wireless Multimedia Sensor Networks," *Computer Networks (Elsevier)*, vol. 51, pp. 921–960, Mar. 2007.

[15] AKYILDIZ, I. F., SU, W., SANKARASUBRAMANIAM, Y., and CAYIRCI, E., "Wireless sensor networks: A survey," *Computer Networks (Elsevier)*, vol. 38, pp. 393–422, March 2002.

[16] ANUJ BATRA ET AL., "Multi-band OFDM Physical Layer Proposal for IEEE 802.15 Task Group 3a." IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs), March 2004.

[17] AURENHAMMER, F., "Voronoi Diagrams - A Survey Of A Fundamental Geometric Data Structure," *ACM Computing Surveys*, vol. 23, pp. 345–405, 1991.

[18] BASHARAT, A., CATBAS, N., and SHAH, M., "A framework for intelligent sensor network with video camera for structural health monitoring of bridges," in *Proc. of the First IEEE Intl. Workshop on Sensor Networks and Systems for Pervasive Computing (PerSeNS 2005)*, (Kauai Island, Hawaii), March 2005.

[19] BETTSTETTER, C., "On the minimum node degree and connectivity of a wireless multihop network," in *in Proc. of ACM Intl. Symp. on Mobile Ad Hoc Networking and Computing (MobiHoc)*, (Lausanne, Switzerland), June 2002.

[20] BHARGHAVAN, V., LU, S., and NANDAGOPAL, T., "Fair queueing in wireless networks: Issues and approaches," *IEEE Personal Communications Magazine*, vol. 6, February 1999.

[21] BOSE, P., MORIN, P., STOJMENOVIC, I., and URRUTIA, J., "Routing with guaranteed delivery in ad hoc wireless networks," *ACM Wireless Networks*, vol. 7, pp. 609–616, November 2001.

[22] BROWN, R. G. and HWANG, P. Y. C., *Introduction to Random Signals and Applied Kalman Filtering, 3rd edition*. Hoboken, NJ: John Wiley & Sons, Inc., 1996.

[23] CAMPBELL, J., GIBBONS, P. B., NATH, S., PILLAI, P., SESHAN, S., and SUKTHANKAR, R., "IrisNet: An Internet-Scale Architecture for Multimedia Sensors," in *Proc. of the ACM Multimedia Conference*, (Singapore), November 2005.

[24] CAO, Y. U., FUKUNAGA, A. S., and KAHNG, A. B., "Cooperative Mobile Robotics: Antecedents and Directions," *Autonomous Robots*, vol. 4, pp. 1–23, 1997.

[25] CHANG, J. H. and TASSIULAS, L., "Energy conserving routing in wireless ad-hoc networks," in *Proc. IEEE Conf. on Computer Communications (INFOCOM)*, pp. 22–31, March 2000.

[26] CHEN, W.-P., HOU, J. C., and SHA, L., "Dynamic Clustering for Acoustic Target Tracking in Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 3, pp. 258–271, July 2004.

[27] CHIANG, M., "Balancing Transport and Physical Layers in Wireless Multihop Networks: Jointly Optimal Congestion Control and Power Control," *IEEE Journal on Selected Areas in Communications*, vol. 23, pp. 104 – 116, January 2005.

[28] CHU, M., REICH, J. E., and ZHAO, F., "Distributed Attention for Large Video Sensor Networks," in *Proc. of the Institute of Defence and Strategic Studies (IDSS)*, (London, UK), February 2004.

[29] CUCCHIARA, R., "Multimedia Surveillance Systems," in *Proc. of ACM Intl. Workshop on Video Surveillance and Sensor Networks* , (Singapore), November 2005.

[30] CUOMO, F., MARTELLO, C., BAIOCCHI, A., and CAPRIOTTI, F., "Radio Resource Sharing for Ad-hoc Networking with UWB," *IEEE Journal on Selected Areas in Communications*, vol. 20, pp. 1722–1732, December 2002.

[31] CZYZYK, J., MESNIER, M., and MORÉ, J., "The NEOS server," *IEEE Journal on Computational Science and Engineering*, vol. 5, pp. 68–75, July-September 1998.

[32] DAS, S. M., PUCHA, H., and HU, Y. C., "Performance Comparison of Scalable Location Services for Geographic Ad Hoc Routing," in *Proceedings of IEEE Conf. on Computer Communications (INFOCOM)*, (Miami, FL, USA), Mar. 2005.

[33] DOWNES, I., RAD, L. B., and AGHAJAN, H., "Development of a Mote for Wireless Image Sensor Networks," in *Proc. of COGnitive systems with Interactive Sensors (COGIS)*, (Paris, France), March 2006.

[34] DURRESI, A., PARUCHURI, V., and BAROLLI, L., "Delay-energy aware routing protocol for sensor and actor networks," in *Proc. of 11th International Conference on Parallel and Distributed Systems*, vol. 1, (Fuduoka, Japan), pp. 292 – 298, July 2005.

[35] EISENBERG, Y., LUNA, C. E., PAPPAS, T. N., BERRY, R., and KATSAGGELOS, A. K., "Joint Source Coding and Transmission Power Management for Energy Efficient Wireless Video Communications," *IEEE Transactions on Cirtuits and Systems for Video Technology*, vol. 12, pp. 411–424, June 2002.

[36] F. KUHN, R. W. and ZOLLINGER, A., "Worst-Case Optimal and Average-Case Efficient Geometric Ad-Hoc Routing," in *Proceedings of ACM Intl. Symp. on Mobile Ad Hoc Networking and Computing (MobiHoc)*, (Annapolis, MD, USA), June 2003.

[37] FELEMBAN, E., LEE, C.-G., EKICI, E., BODER, R., and VURAL, S., "Probabilistic QoS Guarantee in Reliability and Timeliness Domains in Wireless Sensor Networks," in *Proceedings of IEEE Conf. on Computer Communications (INFOCOM)*, (Miami, FL, USA), March 2005.

[38] FENG, W., CODE, B., KAISER, E., SHEA, M., FENG, W., and BAVOIL, L., "Panoptes: Scalable Low-power Video Sensor Networking Technologies," in *Proc. of ACM Multimedia*, (Berkeley, CA, USA), November 2003.

[39] FINN, G., "Routing and addressing problems in large metropolitan-scale internetworks," tech. rep., ISI res. rep ISU/RR- 87-180, March 1987.

[40] FOURER, R., GAY, D. M., and KERNIGHAN, B. W., *AMPL: A Modeling Language for Mathematical Programming*. Duxbury Press / Brooks/Cole Publishing Company, 2002.

[41] GAREY, M. R. and JOHNSON, D. S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY: W. H. Freeman & Co., 1979.

[42] GAY, D., LEVIS, P., VON BEHREN, R., WELSH, M., BREWER, E., and CULLER, D., "The nesC Language: A Holistic Approach to Network Embedded Systems," in *Proc. of the ACM SIGPLAN 2003 Conf. on Programming Language Design and Implementation (PLDI)*, (San Diego, CA, USA), June 2003.

[43] GERKEY, B. P. and MATARIC, M. J., "Sold!: Auction methods for multirobot coordination," *IEEE Transactions on Robotics and Automation*, vol. 18, pp. 758–768, October 2002.

[44] GERKEY, B. P. and MATARIC, M. J., "A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems," *The International Journal of Robotics Research*, vol. 23, pp. 939–954, September 2004.

[45] GEZICI, S., TIAN, Z., GIANNAKIS, G. B., KOBAYASHI, H., MOLISCH, A. F., POOR, H. V., and SAHINOGLU, Z., "Localization via Ultra-Wideband Radios," *IEEE Signal Processing Magazine*, vol. 22, pp. 70–84, July 2005.

[46] GIROD, B., AARON, A., RANE, S., and REBOLLO-MONEDERO, D., "Distributed Video Coding," *Proc. of the IEEE*, vol. 93, pp. 71–83, January 2005.

[47] GIROD, B., KALMAN, M., LIANG, Y., and ZHANG, R., "Advances in Channel-adaptive Video Streaming," *Wireless Communications and Mobile Computing*, vol. 2, pp. 549–552, September 2002.

[48] GOMEZ, J. and CAMPBELL, A. T., "A case for variable-range transmission power control in wireless multihop networks," in *Proceedings of IEEE Conf. on Computer Communications (INFOCOM)*, (Hong Kong S.A.R., P.R. China), March 2004.

[49] GURSES, E. and AKAN, O. B., "Multimedia Communication in Wireless Sensor Networks," *Annals of Telecommunications*, vol. 60, pp. 799–827, July-August 2005.

[50] HE, T., STANKOVIC, J. A., LU, C., and ABDELZAHER, T. F., "A Spatiotemporal Communication Protocol for Wireless Sensor Networks," *IEEE Trans. on Parallel and Distributed Systems*, vol. 16, pp. 995–1006, October 2005.

[51] HEINZELMAN, W., CHANDRAKASAN, A., and BALAKRISHNAN, H., "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, pp. 660–670, October 2002.

[52] HIGHTOWER, J. and BORRIELLO, "Location systems for ubiquitous computing," *IEEE Computer*, vol. 34, pp. 57–66, August 2001.

[53] HU, F. and KUMAR, S., "Multimedia Query with QoS Considerations for Wireless Sensor Networks in Telemedicine," in *Proc. of Society of Photo-Optical Instrumentation Engineers - Intl. Conf. on Internet Multimedia Management Systems*, (Orlando, Florida), September 2003.

[54] HU, L. and EVANS, D., "Localization for mobile sensor networks," in *Proceedings of ACM Intl. Conf. on Mobile Computing and Networking (MobiCom)*, (Philadelphia, PA), September 2004.

[55] JAIN, K., PADHYE, J., PADMANABHAN, V., and QIU, L., "Impact of Interference on Multi-hop Wireless Network Performance," in *Proc. of ACM Intl. Conf. on Mobile Computing and Networking (MobiCom)*, (San Diego, CA, USA), September 2003.

[56] JAMES, D., KLIBANOV, L., TOMPKINS, G., and DIXON-WARREN, S. J., "Inside CMOS Image Sensor Technology." Chipworks White Paper.

[57] JI, X., ZHA, H., METZNER, J., and KESIDIS, G., "Dynamic cluster structure for object detection and tracking in wireless ad-hoc sensor networks," in *Proc. of IEEE Intl. Conf. on Communications (ICC)*, (Paris, France), June 2004.

[58] JIANG, H. and ZHUANG, W., "Effective Packet Scheduling with Fairness Adaptation in Ultra Wideband Wireless Networks," *IEEE Transactions on Wireless Communications*, vol. 6, pp. 680–690, February 2007.

[59] JIANG, X., POLASTRE, J., and CULLER, D., "Perpetual Environmentally Powered Sensor Networks," in *Proc. of IEEE Workshop on Sensor Platform, Tools and Design Methods for Networked Embedded Systems (SPOTS)*, (Los Angeles, CA, USA), April 2005.

[60] KAWADIA, V. and KUMAR, P., "Principles and protocols for power control in ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, pp. 76–88, January 2005.

[61] KUHN, F., MOSCIBRODA, T., and WATTENHOFER, R., "Initializing newly deployed ad hoc and sensor networks," in *Proc. of ACM Intl. Conf. on Mobile Computing and Networking (MobiCom)*, (Philadelphia, PA), September 2004.

[62] KULKARNI, P., GANESAN, D., and SHENOY, P., "The Case for Multi-tier Camera Sensor Network," in *Proc. of the ACM Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, (Stevenson, WA, USA), June 2005.

[63] KULKARNI, P., GANESAN, D., SHENOY, P., and LU, Q., "SensEye: A Multi-tier Camera Sensor Network," in *Proc. of ACM Multimedia*, (Singapore), November 2005.

[64] LATOMBE, J. C., *Robot Motion Planning*. Kluwer, 1990.

[65] LI, J., JANNOTTI, J., COUTO, D. D., KARGER, D., and MORRIS, R., "A scalable location service for geographic ad hoc routing," in *Proc. of ACM/IEEE Intl. Conf. on Mobile Computing and Networking (MobiCom)*, (Boston, Massachussets), pp. 120–30, 2000.

[66] LU, S., NANDAGOPAL, T., and BHARGHAVAN, V., "Design and analysis of an algorithm for fair service in error-prone wireless channels," *ACM Wireless Networks Journal*, vol. 6, January 2000.

[67] MAILLÉ, P. and TUFFIN, B., "Multi-bid auctions for bandwidth allocation in communication networks," in *Proc. of IEEE Conf. on Computer Communications (INFOCOM)*, (Hong Kong S.A.R., P.R. China), March 2004.

[68] MARGI, C. B., PETKOV, V., OBRACZKA, K., and MANDUCHI, R., "Characterizing Energy Consumption in a Visual Sensor Network Testbed," in *Proc. of IEEE/CreateNet Intl. Conf. on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom)*, (Barcelona, Spain), March 2006.

[69] MAYBECK, P. S., *Stochastic models, estimation, and control - Volume 1*. New York: Academic Press, 1979.

[70] MCAFEE, R. P. and MCMILLAN, J., "Auctions and bidding," *Journal of Economic Literature*, vol. 25, pp. 699–738, June 1987.

[71] MCINTIRE, D., "Energy Benefits of 32-bit Microprocessor Wireless Sensing Systems." Sensoria Corporation White Paper.

[72] MEGUERDICHIAN, S., KOUSHANFAR, F., POTKONJAK, M., and SRIVASTAVA, M., "Coverage problems in wireless ad-hoc sensor networks," in *Proc. of IEEE Conf. on Computer Communications (INFOCOM)*, April 2001.

[73] MELODIA, T., POMPILI, D., and AKYILDIZ, I. F., "Optimal local topology knowledge for energy efficient geographical routing in sensor networks," in *Proc. of IEEE Conf. on Computer Communications (INFOCOM)*, (Hong Kong S.A.R., PRC), March 2004.

[74] MELODIA, T., POMPILI, D., and AKYILDIZ, I. F., "On the interdependence of distributed topology control and geographical routing in ad hoc and sensor networks," *Journal of Selected Areas in Communications*, vol. 23, pp. 520–532, March 2005.

[75] MELODIA, T., POMPILI, D., GUNGOR, V. C., and AKYILDIZ, I. F., "A distributed coordination framework for wireless sensor and actor networks," in *Proc. of ACM Intl. Symp. on Mobile Ad Hoc Networking and Computing (MobiHoc)*, (Urbana-Champaign, IL), May 2005.

[76] MERZ, R., WIDMER, J., BOUDEC, J.-Y. L., and RADUNOVIC, B., "A Joint PHY/MAC Architecture for Low-Radiated Power TH-UWB Wireless Ad-Hoc Networks," *Wireless Communications and Mobile Computing Journal*, vol. 5, pp. 567–580, July 2005.

[77] MISRA, S., REISSLEIN, M., and XUE, G., "A Survey of Multimedia Streaming in Wireless Sensor Networks," *submitted for publication*, 2006.

[78] MITCHESON, P. D., GREEN, T. C., YEATMAN, E. M., and HOLMES, A. S., "Architectures for Vibration-Driven Micropower Generators," *Journal of Microelectromechanical Systems*, vol. 13, pp. 429–440, June 2004.

[79] MOLISCH, A. F., BALAKRISHNAN, K., CASSIOLI, D., CHONG, C., EMAMI, S., FORT, A., KAREDAL, J., KUNISCH, J., SCHANTZ, H., SCHUSTER, U., and SIWIAK, K., "A comprehensive model for ultrawideband propagation channels," in *Proc. of IEEE Global Communications Conference (GLOBECOM)*, (Saint Louis, MO), November 2005.

[80] NA, J. and KWON KIM, C., "GLR: A Novel Geographic Routing Scheme for Large Wireless Ad-hoc Networks," *Computer Networks (Elsevier)*, vol. 50, pp. 3434–3448, December 2006.

[81] NATH, S., KE, Y., GIBBONS, P. B., KARP, B., and SESHAN, S., "A Distributed Filtering Architecture for Multimedia Sensors." Intel Research Technical Report IRP-TR-04-16, August 2004.

[82] OGREN, P., FIORELLI, E., and LEONARD, N. E., "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Transactions on Automatic Control*, vol. 49, pp. 1292–1302, Aug. 2004.

[83] PARADISO, J. and STARNER, T., "Energy Scavenging for Mobile and Wireless Electronics," *IEEE Pervasive Computing*, vol. 4, pp. 18–27, January-March 2005.

[84] PAREKH, A. K. and GALLAGER, R. G., "A generalized processor sharing approach to flow control in integrated services networks: The single node case," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 344–357, June 1993.

[85] POTTIE, G. J. and KAISER, W. J., "Wireless integrated network sensors," *Communications of the ACM*, vol. 43, pp. 51–58, May 2000.

[86] RADUNOVIC, B. and BOUDEC, J.-Y. L., "Power Control is Not Required for Wireless Networks in the Linear Regime," in *Proc. of IEEE Intl. Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, (Taormina - Giardini Naxos, Italy), June 2005.

[87] RADUNOVIC, B. and LE BOUDEC, J.-Y., "Optimal Power Control, Scheduling and Routing in UWB Networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 7, p. 1252, 2004.

[88] RAHIMI, M., BAER, R., IROEZI, O., GARCIA, J., WARRIOR, J., ESTRIN, D., and SRIVASTAVA, M., "Cyclops: In Situ Image Sensing and Interpretation in Wireless Sensor Networks," in *Proc. of the ACM Conf. on Embedded Networked Sensor Systems (SenSys)*, (San Diego, CA), November 2005.

[89] RAHIMI, M., AHMADIAN, S., ZATS, D., GARCIA, J., SRIVASTAVA, M., and ESTRIN, D., "Deep Vision: Experiments in Exploiting Vision in Wireless Sensor Networks," in *submitted for publication*, 2006.

[90] RAHIMI, M., SHAH, H., SUKHATME, G., HEIDEMANN, J., and ESTRIN, D., "Studying the feasibility of energy harvesting in a mobile sensor network," in *Proceedings of the IEEE International Conference on Robotics and Automation*, (Taipai, Taiwan), pp. 19–24, May 2003.

[91] REED, J., *Introduction to Ultra Wideband Communication Systems*. Englewood Cliffs, New Jersey: Prentice Hall, June 2005.

[92] REKLEITIS, I., "A Particle Filter Tutorial for Mobile Robot Localization," *Technical Report TR-CIM-04-02, Centre for Intelligent Machines, McGill University, Montreal, Quebec, Canada*, 2004.

[93] ROSSI, M. and ZORZI, M., "Cost Efficient Localized Geographical Forwarding Strategies for Wireless Sensor Networks," in *Proceedings of the Tyrrhenian International Workshop on Digital Communications, TIWDC 2005*, (Sorrento, Italy), July 2005.

[94] ROWE, A., ROSENBERG, C., and NOURBAKHSH, I., "A Low Cost Embedded Color Vision System," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, (Lausanne, Switzerland), October 2002.

[95] SCHAAR, M. V. D. and SHANKAR, S., "Cross-layer Wireless Multimedia Transmission: Challenges, Principles and New Paradigms," *IEEE Wireless Communications*, vol. 12, pp. 50–58, August 2005.

[96] SCHENATO, L., OH, S., and SASTRY, S., "Swarm coordination for pursuit evasion games using sensor networks," in *Proc. of the Intl. Conf. on Robotics and Automation*, (Barcelona, Spain), April 2005.

[97] SCHOLTZ, R. A. and WIN, N. Z., "Impulse Radio: How it Works," *IEEE Communications Letters*, vol. 2, pp. 36–38, February 1998.

[98] SEADA, K., ZUNIGA, M., HELMY, A., and KRISHNAMACHARI, B., "Energy-Efficient Forwarding Strategies for Geographic Routing in Lossy Wireless Sensor Networks," in *Proceedings of IEEE Conf. on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, (Santa Clara, CA, USA), October 2004.

[99] SETTON, E., YOO, T., ZHU, X., GOLDSMITH, A., and GIROD, B., "Cross-Layer Design of Ad Hoc Networks for Real-Time Video Streaming," *IEEE Wireless Communications*, vol. 12, pp. 59–65, August 2005.

[100] SHAH, R., ROY, S., JAIN, S., BRUNETTE, W., and BORRIELLO, G., "Data MULEs: Modeling a Three-tier Architecture for Sparse Sensor Networks," *Ad Hoc Networks (Elsevier)*, vol. 1, pp. 215–233, September 2003.

[101] SHI, Y., HOU, Y., SHERALI, H., and MIDKIFF, S., "Cross-layer optimization for routing data traffic in UWB-based sensor networks," in *Proceedings of ACM Intl. Conf. on Mobile Computing and Networking (MobiCom)*, (Cologne, Germany), August-September 2005.

[102] SORO, S. and HEINZELMAN, W. B., "On the Coverage Problem in Video-based Wireless Sensor Networks," in *Proc. of the IEEE Intl. Conf. on Broadband Communications, Networks, and Systems (BroadNets)*, (Boston, MA, USA), October 2005.

[103] STANKOVIC, J. A., ABDELZAHER, T. F., LU, C., SHA, L., and HOU, J., "Real-time communication and coordination in embedded sensor networks," *Proceedings of the IEEE*, vol. 91, pp. 1002–1022, July 2003.

[104] STOCKDON, H. and HOLMAN, R., "Estimation of Wave Phase Speed and Nearshore Bathymetry from Video Imagery," *Journal of Geophysical Research*, vol. 105, pp. 22,015–22,033, September 2000.

[105] SUNDARARAMAN, B., BUY, U., and KSHEMKALYANI, A., "Clock synchronization for wireless sensor networks: a survey," *Ad Hoc Networks (Elsevier)*, vol. 3, pp. 281–323, May 2005.

[106] VEDANTHAM, R., ZHUANG, Z., and SIVAKUMAR, R., "Hazard Avoidance in Wireless Sensor and Actor Networks," *Computer Communications (Elsevier)*, vol. 29, pp. 2578–2598, August 2006.

[107] VEDANTHAM, R., ZHUANG, Z., and SIVAKUMAR, R., "Mutual Exclusion in Wireless Sensor and Actor Networks," in *Proc of IEEE Conf. on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, (Reston, VA, USA), September 2006.

[108] WANG, G., CAO, G., PORTA, T. L., and ZHANG, W., "Sensor Relocation in Mobile Sensor Networks," in *Proceedings of IEEE Conf. on Computer Communications (INFOCOM)*, (Miami, Fl, USA), March 2005.

[109] WIN, M. Z. and SCHOLTZ, R. A., "Ultra-wide Bandwidth Time-hopping Spread-spectrum Impulse Radio for Wireless Multiple-access Communication," *IEEE Trans. Commun.*, vol. 48, pp. 679–689, April 2000.

[110] YANG, L. and GIANNAKIS, G. B., "Ultra-WideBand Communications: An Idea Whose Time Has Come," *IEEE Signal Processing Magazine*, vol. 3, pp. 26–54, November 2004.

[111] YOUNIS, O. and FAHMY, S., "Distributed clustering in ad-hoc sensor networks: A hybrid, energy-efficient approach," in *Proceedings of IEEE Conf. on Computer Communications (INFOCOM)*, (Hong Kong S.A.R., P.R. China), March 2004.

[112] ZITOVÁ, B. and FLUSSER, J., "Image Registration Methods: A Survey," *Image and Vision Computing*, vol. 21, pp. 977–1000, October 2003.

[113] ZUNIGA, M. and KRISHNAMACHARI, B., "Integrating Future Large Scale Sensor Networks with the Internet." USC Computer Science Technical Report CS 03-792, 2003.

VITA

Tommaso Melodia was born in Rome, Italy. He received the "Laurea" degree in Telecommunications Engineering (equivalent to M.S.) in 2001 from the University of Rome "La Sapienza", Italy.

During 2002 he worked at the same university on the National RAMON project, aimed at designing, developing and demonstrating a reconfigurable module for wireless networking in the context of mobile computing applications.

In 2005, he received his Ph.D. in Information and Communication Engineering from the University of Rome, La Sapienza. Since January 2004, he has been working at the Broadband and Wireless Networking Laboratory (BWN-Lab), Georgia Institute of Technology, Atlanta, under the supervision of Dr. Ian F. Akyildiz. His main research interests are in wireless multimedia sensor and actor networks, underwater acoustic sensor networks, and wireless networks in general.

In August 2007 he will join the Electrical Engineering Department at the State University of New York (SUNY) at Buffalo as an Assistant Professor.